UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

MASTER THESIS

Aspect-based sentiment analysis and item recommendation

Máster Universitario en Investigación e Innovación en Tecnologías de la Información y las Comunicaciones

Author: HERNÁNDEZ RUBIO, María
Tutor: CANTADOR GUTIÉRREZ, Iván

September 2017

Contents

List of figures

List of tables

Glossary

CB *Content-Based*

Content-based recommender systems aim to suggest a user items that are similar to those she liked in the past. For such purpose, they usually utilize user and item profiles composed of content-based features, such as keywords, categories and concepts.

CF *Collaborative Filtering*

Collaborative filtering systems aim to suggest a user items highly rated by like-minded people. Thus, they make predictions (filtering) about a user's preferences by collecting and exploiting preference data from many users (collaborating).

kNN *k-Nearest Neighbors*

Heuristic collaborative filtering technique that uses similarities between users (user-based kNN) or between items (item-based kNN) to perform recommendations.

LDA *Latent Dirichlet allocation*

Generative statistical model that allows sets of observations to be explained by means of latent semantic factors grouping parts of input data that are similar.

MF *Matrix Factorization*

Collaborative filtering technique based on the factorization of a rating matrix into a product of (user and item) latent rating matrices.

NLP *Natural Language Processing*

Computer Science field that intersects artificial intelligence and computational linguistics, and which is concerned with the interactions between computers and human (natural) languages, and, in particular, with programming computers to fruitfully process large natural language corpora.

POS *Part Of Speech*

A category to which a word is assigned in accordance with its syntactic functions, e.g., noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection.

RS *Recommender System*

Information filtering system that aims to retrieve information items relevant to a target user without the need of explicitly stating search queries.

SGD *Stochastic Gradient Descent*

Optimization algorithm in which steps are proportional to the negative gradient of the function. At each iteration, only one (or a few) point is randomly selected to compute the gradient.

Abstract

Recommender systems are software tools that help users to obtain a list of items (i.e., products, services, people, etc.) as answer to (implicit) information needs. In such systems, user and item attributes, as well as user past behavior, are used to estimate a user's preferences and hence provide her with personalized suggestions of items of potential relevance.

In this context, there are many domains –such as restaurants, hotels and e-commerce– where users usually rate available items, and provide textual reviews supporting their ratings. These reviews are a very useful source of information about the user preferences. In particular, the opinions (sentiments) the user has about specific aspects (features, components, etc.) of the items can be exploited to improve the quality of her profile. Recommender systems that use such aspect information are called aspect-based recommender systems.

In this master thesis, we address the aspect-based recommendation task as a three-stage problem. Firstly, performing an *aspect extraction* process where potential item aspects are identified in textual reviews. Secondly, estimating the *opinion* about each aspect a user has commented on. Finally, exploiting the obtained aspects and their opinion polarities (i.e., positive, neutral, and negative) to generate effective personalized *recommendations*.

For the aspect extraction task, we have developed and empirically compared two state-of-the-art, popular methods, namely *Double Propagation* –which exploits semantic relations between the words in the review to establish which of such words may correspond to aspects–, and *Topic Models* –which find latent topics as a proxy for the aspects. Next, for the aspect opinion polarity estimation stage, we have followed the common strategy of using a lexicon –i.e., a list of well-known words that are positive or negative in general domains–, but differently to previous work, we have used Natural Language Processing techniques and resources to better estimate the opinion polarity in negated adjectives and negative sentences. Finally, for the aspect-recommendation stage, we have implemented and evaluated numerous recommenders of several types, such as content-based, collaborative filtering, and hybrid, with and without using aspect-based information.

We have conducted an exhaustive experimentation on several domains with relatively large datasets, and computing a wide array of metrics. The obtained results show that considering the opinion about item aspects generates valuable recommendations that improve the performance of personalized recommendation methods, and have empirically proved that content-based recommendation approaches with an appropriate aspect-based user representation achieves the best performance results.

x

Acknowledgements

This thesis is the result of more than a year of dedicated and exhaustive work.

I would firstly like to thank my supervisor, Dr. Iván Cantador, for his extraordinary help and support over this time. His experience and expertise in the field have been very valuable to this work.

I would also like to acknowledge Dr. Ignacio Fernández and Dr. Alejandro Bellogín for their help to understand and implement recommendation methods that are presented here. They have been very helpful with me in this time as well.

Thanks to my colleagues and friends, the Data-team, for their daily support. They not only have helped to serenade me, but also encouraged me in the final stages. Thanks to my friends, even though they asked me what my thesis topic was every time we saw each other. Thanks also for some of their feedback and interesting questions that helped me to guide my research.

I would also like to thank family, my parents and my brother Alvarito, for their support and confidence they have always placed on me.

And last but not least I thank my partner, for being able to understand and assume that *he didn't have a girlfriend* during the last months.

Chapter 1

Introduction

In this master thesis we aim to develop and evaluate a number of solutions to the aspect-based recommendation task, which consists of providing a user with personalized suggestions of items taking into account her and others' opinions about particular aspects of existing items. For such purpose, we address three research topics, namely the identification of aspects discussed in textual reviews, the polarity classification of aspect opinions in such reviews, and the exploitation of extracted aspect opinions to generate effective item recommendations.

In Section 1.1 we motivate the thesis, by discussing the relevance and challenges of aspect-based sentiment analysis and aspect-based item recommender systems. Next, in Section 1.2 we describe the main goals of our research, and in Section 0 we list our contributions. Finally, in Section 1.5 we provide the structure of the thesis, summarizing the contents of its chapters.

1.1 Motivation

The huge, ever-increasing growth of the Web has led to a large number of applications that have been migrated from the physical to the digital world. Markets such as shopping, banking, multimedia consumption, and service booking, to name a few, are example cases where daily activities are done online.

This shift has changed the way humans consume information. In the physical world, we usually search for information on a limited number of resources; any information need could be solved by checking, at most, a few items. For instance, let us suppose we want to cook a dish for the first time. In this case, we may search for receipts in several cooking books available in our library, read two or three recipes, and select and follow the one we have considered as the most suitable. Another example: buying

a mobile phone. In this case, we may go to several electronic shops, and ask assistants for information about phone characteristics and recommendations about the best phone for us. Again, after a few consultations, we would make a choice.

In the digital world, in contrast, the existing resources and amount of information are so large that is unfeasible to manually check and compare all available options. A simple "recipes for summer dinner" query returns more than 37 million results on Google[1]. In this scenario, we need computer-assisted mechanisms that help us finding the best solution to our needs in an efficient, simple way.

The information overload problem is aggravated considering that the majority of online contents are unstructured texts, and we have to ask for answers by means of keyword-based queries. In this scenario, it would desirable to have search engines able to understand the semantics underlying free text contents, in order to identify the Web resources that provide the adequate answers to particular questions.

Online reviews are an important asset for users who have to decide among options, e.g., for buying certain type of product, watching a movie, or going to a restaurant (Ganu et al., 2009). Trough reviews users express their opinions about a wide array of items and their "aspects", i.e., characteristics, features, attributes or components. Reviews are usually written in a free text format, and users need to carefully read them to identify the expressed opinions, and find out the strengths and weaknesses of the available items, for making the best decisions.

Addressing the information overload problem, and helping users in decision making tasks, recommender systems aim to provide the users with personalized suggestions of the most valuable items. In general, these systems exploit (numeric) ratings that users assign to items, and recommend to a target user the items that are most similar to those she liked in the past, and the items preferred by like-minded people.

In addition to ratings, recommender systems could also analyze and use the opinions expressed in textual reviews. Ratings act as summaries of the users' preferences, and do not reflect the details about their opinions usually expressed in the reviews. Exploiting these textual contents may allow a recommender system to better understand both user preferences and item aspects, and generate more useful and informed recommendations.

For instance, let us consider a user who rates a mobile phone with an overall rating of 4 stars. With no more information, it is not possible to know why she gave such score. Analyzing a review she wrote about the phone, we may know that the user thinks the phone camera is the best she has used, and that the life of the phone battery is long,

[1] Query launched on September 2017.

close to almost two days. Moreover, we may discover that the user find the phone a bit heavy and quite expensive. These opinions about particular aspects of the phone are the reasons for the user's 4-stars rating. Considering only the numeric value of the rating, a recommender system would treat this phone as worse than any other with a 5-stars rating.

The above facts represent a summary of the user's opinion about the phone, and are focused on particular item aspects. Considering these aspects, Table 1.1 shows a possible precise representation of the user's opinion, composed by aspect ratings. Textual reviews, in contrast, are usually much less schematic and contain sentences and words that make the automatic identification of aspect opinions a challenging task. Hence, in this thesis, we aim to develop and evaluate methods for **extracting the item aspects discussed in user reviews, and classifying the polarity (i.e., positive, neutral and negative) of the given opinions on extracted aspects**.

Aspect	**Rating**
Camera	5
Battery life	5
Weight	4
Price	3

Table 1.1 Example of an aspect-based review representation.

Following the previous example, let us think about another user who is interested in buying a phone, and does not care about its price. For this user, the last aspect of the given review is indifferent, and discarding it, the reviewed phone would have a rating higher than 4 stars. Providing personalized item recommendations by taking into consideration opinions on particular item aspects is also a difficult task, and represents the second research challenge we address in this thesis. For such purpose, we aim to develop and evaluate state-of-the-art and novel methods for **aspect-based recommendations**.

More specifically, in the research literature, the previous tasks have been called as *aspect-based sentiment analysis* and *aspect-based recommendation*, respectively. In the former, proposed solutions aim to analyze user textual reviews, and extract the items aspects that are being discussed, together with their opinion polarities. In the latter, existing approaches aim to exploit aspect opinion polarities to improve personalized item recommendations, by means of natural language processing, machine learning, and collaborative filtering techniques.

1.2 Goals

A first goal of this master thesis is to provide a review of the state of the art in aspect extraction and sentiment analysis as well as aspect-based recommender systems. For such purpose, we shall provide a formulation of the problem, describe and compare relevant works in the Opinion Mining and Recommender Systems fields, and identify which of the existing approaches are the most effective and relevant for being further investigated.

Our second goal is the implementation of the selected approaches to aspect extraction, aspect opinion polarity classification, and aspect-based recommendation. In this case, we also aim to develop adaptations of existing approaches and propose new approaches.

Finally, a third goal is an exhaustive offline evaluation of the previous methods using large, public datasets on several application domains. With the conducted experiments, we aim to determine which solutions are the most effective, and under which circumstances.

1.3 Research questions

As a result of the stated goals, we aim to address the following research questions:

- **RQ1: Are opinions about item aspects in user reviews valuable to improve the performance of personalized recommendation methods?** To address this question, we shall compare several well-known recommendation algorithms with and without using aspect opinion data extracted from real reviews in several domains.
- **RQ2: Which approach to aspect extraction generates the most valuable aspect-based information for recommendation purposes?** To address this question we shall evaluate two popular, state-of-the-art methods to aspect extraction, namely the Double Propagation algorithm, and a Topic Models.
- **RQ3: Which recommendation technique takes more benefit of aspect-based information?** To address this question, we shall evaluate several methods generating different types of recommendations, namely content-based, collaborative filtering, and hybrid recommendations.

1.4 Contributions

As a result of the conducted review of the research literature on aspect-based sentiment analysis and recommendation, we have proposed a novel categorization of existing approaches. This has allowed us to properly evaluate several combinations of methods for each of the involved tasks, namely aspect extraction, aspect opinion polarity classification, and aspect-based recommendation.

We have implemented and evaluated a significant number of methods for the above tasks, with relatively large public datasets on several application domains. In this context, we have developed user and item representations in terms of relevant aspects and exploited them in content-based, collaborative filtering and hybrid recommendation methods. We show that the use of aspect-based information improve the performance of personalized recommendation methods.

Finally, in the conducted evaluation, we have analyzed how the outputs of existing aspect extraction methods affect the performance of subsequent aspect-based recommendations.

1.5 Structure of the thesis

This thesis is structured as follows:

- In **Chapter 2** we present an overview of the Sentiment Analysis and Recommender Systems fields, in order to provide the reader with background knowledge needed to understand the addressed tasks and developed solutions.
- In **Chapter 3** we revise the research literature, describing and discussing state-of-the-art approaches for the abovementioned tasks, and pointing out their main strengths and weaknesses or limitations.
- In **Chapter 4** we present the methods we have developed to address the target tasks. In particular, in Section 4.1 we present the methods for aspect extraction and sentiment analysis, and in Section 4.2 we present the methods for aspect-based recommendation.
- In **Chapter 5** we report and analyze the results achieved in the experiments conducted to evaluate the developed methods.
- In **Chapter 6** we present some conclusions of our work, as well as potential research lines that may be addressed in the future as a continuation of this thesis.

Chapter 2

Background and context

In this chapter we provide some background on the research areas and topics related to this thesis. Giving a general overview of the state-of-the-art on such areas and topics, we aim to allow the reader gaining an easier and better understanding of our work.

Specifically, in Section 2.1 we give definitions and main issues of sentiment analysis and opinion mining, focusing on aspect-based sentiment analysis; and in Section 0 we discuss recommender systems, putting special emphasis on aspect-based recommender systems, and standard metrics utilized to measure the quality of recommendations.

2.1 Sentiment analysis and opinion mining

Broadly, **sentiment analysis** refers to computational processes for the automatic identification and classification of the sentiments or opinions someone has about an entity/item (e.g., a product, a person, an event, an organization) (Liu, 2012). This task is usually conducted on text reviews and speech transcriptions, but has addressed in other forms, such as emotion recognition in facial expressions. Hence, sentiment analysis may involve the use of a variety of natural language processing, text analysis, computational linguistics, and biometrics techniques.

Sentiment analysis is usually called as **opinion mining**. The differences between them are very subtle and are often ignored. Sentiment analysis refers more to the *internal* feeling a person has about an entity, whereas opinion mining is concerned about extracting the sentiment a person has *expressed* about an entity. Both concepts are used indistinctly, but it has to be noted that person might have an opinion about a particular entity, and may not have entirely expressed it.

The explosive growth of the Web has led to a huge amount of recorded opinionated data in digital forms. In this scenario, opinions are found in a wide variety of text information sources, such as blogs, social networks, product reviews, and photo and video comments, among others. Hence, Opinion Mining has been included in the Natural Language Processing (NLP) community as a very important research area.

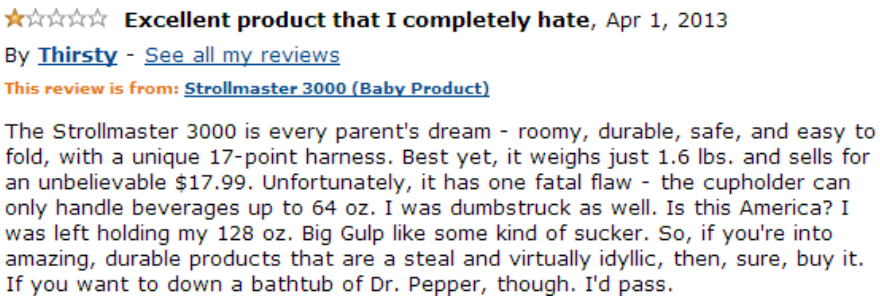

Figure 2.1 Example of Amazon review about a product.

Moreover, there is a wide range of applications where sentiment analysis is a core component. For instance, companies are interested in knowing what consumers think about their products. For such purpose, they analyze the users' reviews in the web to improve their products and sale offers. Consumers, on the other hand, usually read textual reviews about products before purchasing. Other users' opinions may be very helpful to get a better understanding on the strengths and weaknesses of each product. Providing computer-assisted solutions to automatically address or support these tasks are thus highly valuable.

2.1.1 Opinion definition and opinion mining tasks

Opinions expressed in the form of textual reviews share some common elements that correspond to the key parts of an opinion, namely the *opinion target* and the *opinion polarity*.

- The **opinion target** is the entity on which an opinion has been expressed. For example, the sentence "I find this mp3 player really useful" expresses a sentiment about the entity *mp3 player*. The entity target may be a product, a person, an organization or an event, among others.
- In its simplest form, the **opinion polarity** can be positive or negative. In the previous example, the author expresses a *positive* sentiment about the mp3 player. In contrast, the sentence "I don't recommend to buy this TV" represents a *negative* sentiment about certain *TV*. A sentiment can also be neutral if the user does not express polarity about the item she is talking about, as in the sentence "I bought this book 3 years ago", where there is neither explicit nor implicit opinions about the *book*.

In addition, the user can also express a relative degree of sentiment she has on the entity. For example, "This mp3 is the best I've ever had" expresses a higher positive sentiment about the mp3 than "This mp3 works quite well." Being able to capture this sentiment degree is also an objective of opinion mining tasks.

Besides the above two key elements, there are other components that can be found in an opinion:

- The ***opinion holder*** is the person who makes the opinion. This person may or may not correspond to the author of the text. For example, in the sentence "My sister thinks this mp3 player is the best she has ever had," the opinion holder is the author's sister.
- The ***time*** (and date) of the opinion, which does not need to coincide with the time when the review is written.
- The ***target aspect*** on which the opinion is expressed, where 'aspect' is a component, part or feature of the target entity. In the examples given above, all the opinions refer to the target as a whole. In these cases, we can consider that the aspect is *GENERAL*. Differently, there are many opinions that are about particular aspects. For example, let us consider the sentences "The camera of this mobile is very good. However, the battery life is very short." The user is expressing a *positive* opinion about the *camera* of a mobile phone, and a *negative* one about its *battery*. Both, *camera* and *battery* are components of the entity *mobile*, and the user is not expressing a global opinion about the entity.

These elements lead to the formal definition of ***opinion*** as a 5-tuple (Liu, 2012),

$$(e, a, s, h, t)$$

where e is the target entity, a is the target aspect of entity e on which the opinion has been given, s is the sentiment of the opinion on aspect a of entity e, h is the opinion holder, and t is the opinion posting time.

Figure 2.2 shows review example about The Lawnmower Man from (Wang et al., 2012), where we can observe some of the 5 components of an opinion. It is a review about the entity The Lawnmower Man, on time 2017, August 12th, and the opinion holder is not shown in the fragment. There are four aspects (underlined) whose sentiments are all positive, except the last one.

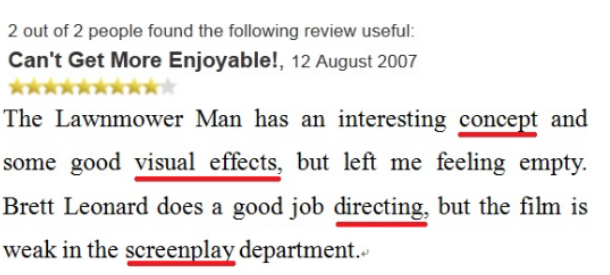
2 out of 2 people found the following review useful:
Can't Get More Enjoyable!, 12 August 2007
The Lawnmower Man has an interesting concept and some good visual effects, but left me feeling empty. Brett Leonard does a good job directing, but the film is weak in the screenplay department.

Figure 2.2 A review example about *The Lawnmower Man* from (Wang et al., 2012).

In general, opinion mining tasks aim to first extract the opinion tuples from text reviews. It could be the case, nonetheless, that not all of the tuple elements are expressed for a particular entity. They may not be available or they may be implicit, e.g., the aspect "price" is implicit in "iPhone7 has a great camera and a long battery duration, however it is very expensive".

As discussed in (Liu, 2012), there are other types of opinions that do not fit in the schema given before. For instance, *comparative opinions* compare a sentiment on an entity with respect to another entity based on some shared aspects.

Moreover, the given definition is valid for a 'regular opinion', that is, an opinion that expresses a sentiment about a single entity. Regular opinions can be further divided into *direct* and *indirect opinions*. Direct opinions are those where the target is the main entity referred in a sentence. Indirect opinions, in contrast, are those where the target is another entity, which is a consequence or is related to the main entity. For example, the sentence "After I read the book, I understand South America history much better" provides an indirect opinion about the *book*. Most of sentiment analysis research focuses on direct regular opinions since other types of opinions are more difficult to handle. We will do so in this work, and refer to them simply as *opinions*.

2.1.2 Sentiment analysis research issues

As already mentioned, sentiment analysis aims to first extract and analyze the 5 components of the opinion tuple defined in Section 2.1.1. The target entity, holder, and posting time are usually easy to obtain, since they are generally explicitly included in the texts metadata. Thus, in general, the focus is at obtaining the opinions and their associated polarities.

Extracting the opinions

Sentiment classification (on textual data) is a well studied problem (Wiebe, 2000, Pang et al., 2002; Turney, 2002), and is carried out at three levels of granularity, namely *document level*, *sentence level*, and *aspect level*.

- At the **document level**, the goal is to determine whether a whole document expresses a positive, negative or neutral opinion, assuming a document only contains a general opinion on a single entity.

- At the **sentence level**, the opinion orientation of each sentence is analyzed independently. In this case, a sentence is first classified into objective or subjective. Next, objective sentences are established as neutral opinion, and subjective sentences are further classified into positive or negative opinions.
- Finally, at the **aspect level**, the objective is to find the opinion the holder has about each 'aspect' of the target entity. An aspect is usually referenced with a word or a (small) set of words, which are the names of the aspects of the entity. Sentiment classification at this level allows for the extraction of different sentiments for several aspects of an entity, and let understand which are the differences that make a user preferring one item more than another.

Determining the opinion polarity

Several studies (Hu and Liu, 2004a; Qiu et al., 2011) have observed that opinion is mainly expressed with *adjectives*, followed by verbs and compound expressions. This observation is the main assumption of a very fruitful research area where opinions are constructed based on the adjectives found in the texts. However, considering only adjective words is not enough, and there are other problems that arise when analyzing opinion polarities in textual data:

- A particular word can be positive or negative depending on the domain or the context. For example, consider the adjective *high*. When it refers to the life duration of a battery, it has a positive orientation, whereas referring to the price of a battery, it has a negative connotation.
- Sarcasm, i.e., double sense meaning, could completely change the primary meaning of a sentence, for example "What a great car! It stopped working in two days". This is one of the hardest issues to detect in sentiment analysis.
- There are sentences that contain opinion words, but do not express an opinion, such as interrogative or conditional sentences, e.g., "Do you consider the new house Mary bought is beautiful and well located?" Moreover, there are sentences that express an opinion, but contain no opinion words, as in "Every time I want to watch a DVD on it, I need to try it twice".
- Finally, negations play a key role in opinion mining since they make the polarity of an opinion word to be the opposite. For example, the sentence "I don't find this manual very useful" express a negative opinion about the manual although the word useful is positive. Negations have to be carefully analyzed in order to identify which parts of the sentences are affected.

These and other types of considerations make the sentiment analysis problem quite subtle, even though it may seem reasonable well defined.

2.1.3 Aspect-based sentiment analysis

As mentioned before, the goal of aspect-based sentiment analysis is to identify the item aspects that are being discussed in textual reviews, and classify them according to the sentiment the author has about them. This multi-objective process can be addressed jointly or through different stages.

Aspect extraction

Aspect extraction was first addressed in the user review summarization task (Hu and Liu, 2004a, 2004b; Popescu and Etzioni, 2005; Zhuang et al., 2006; McAuley et al., 2012), where the goal is to identify the item features that are being discussed in reviews about certain item, and use such features to generate a summary of the reviews by means of a few short sentences.

Within the context of extracting item aspects from user reviews, aspects can be classified as *implicit*, when they are not mentioned, but indirectly referenced in an input review, and *explicit*, if they are cited in the review. For instance, in the sentence "this car is expensive," 'expensive' is a sentiment word that refers to the implicit aspect 'price', whereas in the sentence "this car has a very high price," the aspect 'price' is referred explicitly, and is accompanied with the sentiment word 'high.' Although there are works on the extraction of implicit aspects such as (Popescu and Etzioni, 2005; Poria et al., 2014), most of published researches have focused on identifying explicit aspects. For this task, four main approaches can be found in the literature.

A first approach is based on the exploitation of **frequencies of words** expressing item aspects in certain domain (Scaffidi et al., 2007). Assuming that aspects are nouns or noun phrases, those that appear at a high ratio in a collection of related reviews are considered as candidates to be aspects. For instance, 'soundtrack' and 'astringent' usually appear in movie and wine reviews respectively much more often than in generic, multi-domain text corpora. A method based on this idea (Caputo et al., 2017) has been used recently by considering the divergence between (aspect) words appearance distributions in the target domain with respect to their distributions in a multi-domain, generic corpus.

A second approach focuses on the exploitation of the **syntactic relations** existing between nouns and adjectives in the reviews sentences. When an adjective expresses an opinion, the word that it modifies is a candidate aspect, as in "the new iPhone 8 has a terrific camera", where the adjective 'terrific' modifies the aspect 'camera'. In this context, if certain noun has already been identified as an aspect, syntactically related nouns are candidate words for describing aspects, e.g., 'script' in the sentence "the photography and script are the best in this movie!" if 'photography' is known to be a movie aspect. A popular algorithm that follows this approach is Double Propagation (Qiu et al., 2011), which will be presented in detail in Section 3.1.

A third approach consists on building **supervised learning** models for information extraction. Sequential learning methods such as Hidden Markov Models (HMM) (Rabiner, 1989) and Conditional Random fields (CRF) (Lafferty et al., 2001) are some of the proposed techniques to address the aspect extraction task (Jakob and Gurevych, 2010; Jin and Ho, 2009). These methods label sequences of words based on hidden state sequences. In aspect extraction, words and phrases of a review are treated as tokens, and opinion expressions as underlying states. Training data is annotated with (target token, opinion) pairs, and model parameters are learnt to maximize the probability that input review sentences (i.e., token sequences) have associated opinions.

The previous three approaches share a common problem: people may use different words to refer to a particular concept, and thus an aspect is usually referred with several words. In order to overcome this problem, extracted aspect words are usually grouped together, usually by means of lexicographical similarities, synonym relationships, and taxonomy-based distances (Carenini et al., 2005).

Addressing this problem, a fourth approach has been investigated that uses **topic models** to simultaneously extract and group aspects. Topic models receive a set of documents, and identify the topics (and their distributions) of the documents. In this approach, the topics are usually composed of a set of words, and a topic distribution indicates the 'proportion' of a document that discusses the topic. For aspect extraction, reviews are treated as documents, and topics may represent the aspects the reviews talk about. A topic thus can be seen as an aspect category that is represented as a number of words describing the aspect. In this context, the approach is able to find both explicit and implicit aspects. For instance, words like 'price', 'cheap', 'expensive' and 'unaffordable' may be grouped in the same topic, i.e., price. The majority of topic models for aspect extraction and sentiment analysis are based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and probabilistic Latent Semantic Analysis (pLSA) (Hofmann, 2001), which exploit word co-occurrences within documents and word distribution differences to infer semantic clusters (topics) for the collection. Reviews about products in certain domain may contain opinions about a limited set of aspects, and thus their topic distributions may be very similar. For this reason, topic models for aspect extraction usually extend global topic models (Diao et al., 2014; Titov and McDonald, 2008a; Zhao et al., 2010) to overcome this issue and find more suitable aspects. In this context, a popular method is to differentiate word categories in the generation process of a topic model, such as *global aspect word*, *specific aspect word*, *opinion word* and *background word*. Separating those word types forces the generation process to extract the most useful information.

Aspect sentiment classification

Applications not only need to know the aspects the reviews talk about, but also the sentiments expressed about such aspects. We can distinguish between two main

approaches for the sentiment classification stage: *supervised* and *lexicon-based* sentiment classification approaches.

In its basic form, a **lexicon** is a list containing positive and negative words or expressions. Ideally, this list should be associated to a particular context, since certain word may have different polarities in different contexts (see Section 2.1.1 above). It is usually the result of an automatic construction procedure. This process, in general, starts with a seed list of known (general-purpose) sentiment words (e.g., great, good, excellent, fantastic, bad, nasty, poor, wrong, awful). Then, the list is expanded using synonyms and antonyms from a dictionary (e.g., WordNet (Miller, 1995)). Rules that relate different terms could be used as well. For example, if two terms appear in the same context linked through the conjunction "and", they are considered to have the same polarity. Contrarily, those linked through connector "but" are considered to be opposite. These restrictions are called *sentiment consistency*.

Several researches have led to the construction of general-purpose lexicons, and some of them are publicly available, as Sentiment Lexicon (Hu and Liu, 2004b) and SentiWordNet (Esuli and Sebastiani, 2006).

In l*exicon-based* aspect sentiment classification, we start with a well-known lexicon, find the words and expressions in the text reviews, and then identify the target words that they affect (Hu and Liu, 2004b).

Syntactic relations and typed dependencies are commonly used. If a target word is in the scope of a lexicon word, its polarity (+1 if positive, –1 if negative) is considered in the aspect global sentiment, which can be then computed as a function of the polarities of the opinion words that affect the target. The quality of this unsupervised approach is based on the soundness of the lexicon and the syntactic analysis quality.

Some of the main issues in sentiment analysis that we have shown above need to be considered in this stage, such as negations and *but*-clauses. Besides, opinion intensifiers may affect the degree of the positive or negative opinions. Adverbs such as very, incredibly or really, emphasizes the opinion about the aspect.

Supervised approaches make use of the item ratings to infer each item aspect polarity (Wang and Blei, 2011; Bauman et al., 2016). It assumes that item aspects that appear in high-rated reviews have a more positive opinion than those that are named in negative reviews.

We will describe some aspect extraction procedures and sentiment classification in detail in the Related Work at Section 3.1.

2.2 Recommender systems

Recommender Systems (RS) are information filtering systems that take a user' preferences –i.e., tastes, interests and needs– into account to select those items (e.g., products, movies, music albums, people, etc.) which could be the most "relevant" for the user.

They are usually applied in situations where the collection of available items is so large that overwhelms the user's search capabilities. In such situations, when a user is looking for particular items –either by browsing category taxonomies or by launching keyword-based queries–, she is presented with a result list that may contain dozens, even hundreds of items. Then, the user has to carefully explore the list, and select the items that she likes the most. This may lead her to quit the process, since she has to spend too much time to find suitable items that fits her information needs.

Recommender systems aim to perform or assist this searching process, so that items that the user may like the most are placed at the beginning of ranking lists. The underlying task is thus to filter and sort the collection of items according to the user's preferences. The sorting can be done through different algorithms, considering distinct signals of information about the users, the items, and the user's current context.

There are many domains where recommender systems play a fundamental role. E-commerce is likely the domain that has taken more benefit from recommendation solutions, offering personalized suggestions of a wide array of items, including books, music, electronic devices, and clothes, to name a few. Figure 2.3 shows Amazon[2] website, which suggests the user items that are similar to the selected book since it is very likely that she will also like them. Other domains where RS are frequently applied are hotel booking, on-demand media streaming, digital music and online dating[3], among others.

Recommender systems use past data about the user to infer what she will like the most in the present or future. Past user preferences can be *explicit* or *implicit*.

[2] http://www.amazon.com

[3] http://www.booking.com, http://www.netflix.com, http://www.spotify.com, http://www.match.com

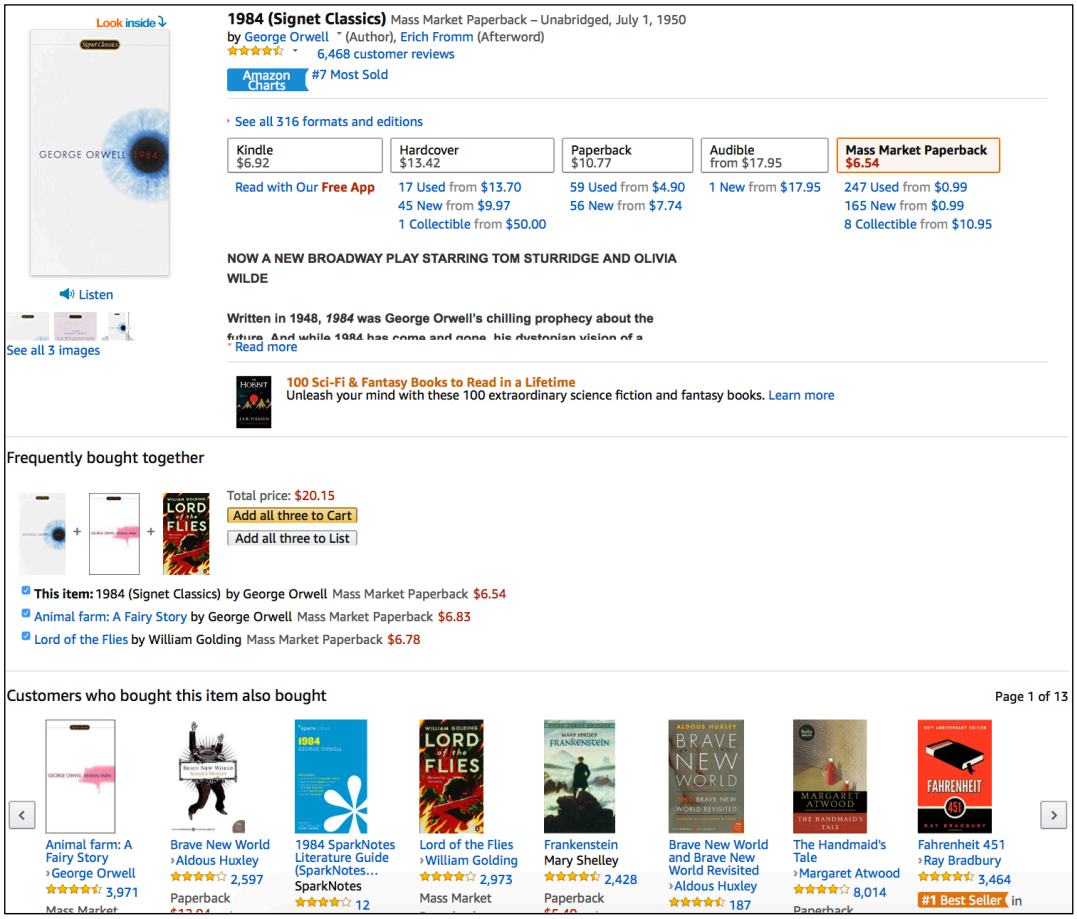

Figure 2.3 Examples of Amazon website recommendations based on those items that are usually bought together with a selected book.

Explicit feedback is that the user is asked and explicitly rates an item. It is usually in the form of a yes/no vote, as the *thumbs up/down* in YouTube and the *likes* on Facebook[4]; or a numeric rating –usually from a valid range of stars–, as in Amazon and FilmAffinity[5].

Implicit feedback is not directly provided by the user but inferred from the behavior within the system (Nichols, 1998). For example, in a streaming service of digital music, a user that only listens to certain song a few seconds and then switches to the next song is likely that she does not like it; whereas if she listens to it several times, she will probably give it a very high rating. If she also listens to several songs from the same author, it is likely that she likes that author. Clicks, page views, purchase actions or time spent are some types of implicit feedback sources (Oard and Kim, 1998).

[4] http://www.youtube.com, http://www.facebook.com

[5] http://www.filmaffinity.com

Most of the literature has focused on explicit feedback, probably because the simplicity of using this type of information, but in recent years research has moved to analyzing implicit feedback, which is the most extended in practice (Hu et al., 2008).

In implicit feedback, the rating is usually considered to be only positive or negative, and intensity in the service use is associated with the confidence about the feedback, not the scale (Hu et al., 2008). For example, a user will see her preferred film a few times at most, whereas a series that barely likes will see it every week.

2.2.1 Types of recommender systems

There are several types of recommender systems, depending on the assumptions and the information they use to estimate the relevance of certain item for a target user.

In particular, let us denote u and i be a user and an item respectively, where u has never expressed a preference opinion about i; and let us consider that user preferences are expressed by means of ratings, explicitly provided or implicitly inferred. The main goal of a recommender system is to estimate the rating $\hat{r}$ that user u would give to item i. The way such rating is estimated leads to different types of recommendations, as presented next.

Content-based recommender systems

In Content-Based (CB) recommender systems, users and items are represented by means of (content) features, and it is assumed that a user will give higher ratings to items that are *similar* to those she liked in the past.

The features used to describe users and items can be discrete or continuous attributes. For instance, in an e-commerce site, common attributes are the size, color and price of products, whereas in an on-demand TV and movie streaming service, typical attributes are the director, actors and duration of films and TV series. Moreover, attributes can be set manually when the items are registered into the system; or can be inferred from their data. For instance, in an e-commerce site, the color of a product could be inferred from its images.

In any case, the content features of an item i are usually represented as a vector $\mathbf{i}$ whose components have associated weights that represent the importance of the features to describe the item. One of the most popular techniques to compute such weights is the well-known TF-IDF (Term frequency-Inverse Document Frequency) score.

This feature-based vector representation is also used for a user's profile $\mathbf{u}$. In this case, the weights are computed by aggregating the corresponding weights of the items the users liked in the past. Thus, a user u's profile is created by combining the profiles of the items $i_k \in I_u$ rated by the user, weighted by the rating $r(u, i_k)$ the user assigned to them:

$$\mathbf{u} = \sum_{i_k \in I_u} r(u,\, i_k)\, \mathbf{i}_k$$

As an illustrative example, if a user has mostly evaluated *action* movies with higher ratings than *comedy* movies, the user's profile will have a higher weight at the *action* component than at the *comedy* component.

For the above feature-based profile representations, the estimated rating of a user u to a new unrated item i is computed by means of a similarity metric between u's and i's vectors:

$$\hat{r}(u, i) = sim(\mathbf{u}, \mathbf{i})$$

Several similarity metrics can be used, most of them based on vector distance metrics. A commonly used metric is the well-known *cosine* similarity:

$$\cos(\mathbf{u}, \mathbf{i}) = \frac{\mathbf{u} \cdot \mathbf{i}}{|\mathbf{u}||\mathbf{i}|}$$

One of the principal drawbacks of content-based recommendation approaches is the so-called *content overspecialization problem*, i.e., suggested items are too similar, and may not offer diversity and novelty. Moreover, as the user's profile is computed from her rating history, she has had to rate several items before she can receive personalized recommendations. This is called as the *user cold-start problem*, and it is something most of recommendation algorithms suffer from. Finally, users may change their tastes and interests over time, so ratings provided long time ago may be no longer valuable to estimate new ratings.

Collaborative filtering

Differently to CB strategies, Collaborative Filtering (CF) systems consider the preferences of like-minded people to estimate a user's ratings. This methodology exploits the available information about other users' ratings *collaboratively* instead of only using the target user's ratings. The assumption here is that if two users have similar preferences on certain items, they are likely to have a more similar preference on different items than two random users.

Since CF only uses previous ratings, and not content-based features, they just need triples (user, item, rating) as source of information, which allows considering items from different types and domains.

CF methods can be further divided into memory- and model-based methods.

- **Memory-based collaborative filtering**

  These methods are also called *kNN* (k nearest-neighbors) and *heuristic* methods. Their core idea is considering a limited number of similar users (i.e., *neighbors*), and exploit their ratings for the target item (**user-based CF**) (Shardanand and Maes, 1995); or analogously considering a limited number of items similarly rated

to the target item (**item-based CF**) (Sarwar et al., 2001) and exploit the ratings the user has assigned them. More formally, the estimation of ratings $\hat{r}(u,i)$ is done as follows.

- *User-based CF*, or *UB-CF*, recommends to u the items highly rated by like-minded people. For such purpose, it takes the n most "similar" users to u,

$$u_k \in \eta_n(u), k = 1 \dots n$$

and aggregates the ratings that u_k assigned to a target item i, by weighting them with the similarity between u and u_k:

$$\hat{r}(u,i) = c \sum_{u_k \in \eta_n(u)} r(u_k,i) \cdot sim(u,u_k)$$

where c is a normalization factor so that estimated rating is in the same scale as the existing ratings.

Cosine distance and Pearson's correlation are the typical metrics to measure the similarity between users. They are computed with the ratings assigned to the items that are rated by both users, $I = I(u) \cap I(u_k)$:

$$sim(u,u_k) = cos\big(r(u), r(u_k)\big) = \frac{\sum_{j \in I} r(u,j) r(u_k,j)}{\sqrt{\sum_{j \in I} r(u,j)^2} \sqrt{\sum_{j \in I} r(u_k,j)^2}}$$

$$sim(u,u_k) = Pearson\big(r(u), r(u_k)\big) = \frac{\sum_{j \in I} \big(r(u,j) - \bar{r}(u)\big)\big(r(u_k,j) - \bar{r}(u_k)\big)}{\sqrt{\sum_{j \in I} \big(r(u,j) - \bar{r}(u)\big)^2} \sqrt{\sum_{j \in I} \big(r(u_k,j) - \bar{r}(u_k)\big)^2}}$$

- *Item-based CF*, or *IB-CF*, recommends to u the items that are most "similar" to the items highly rated by u. For such purpose, it takes the top n items $i_k \in I_n(u)$ rated by user u, and computes the estimated rating as a combination of the similarities between i and i_k:

$$\hat{r}(u,i) = c \sum_{i_k \in I_n(u)} r(u,i_k) \cdot sim(i,i_k)$$

In this case, the similarity between items is computed as the similarity between the vectors of ratings assigned by users that have rated both items, $U = U(i) \cap U(i_k)$.

The performance of these methods is sensible to the number of neighbors considered to estimate the ratings, which has to be set empirically for the dataset used.

- **Model-based collaborative filtering**

The kNN methods are based of heuristic formulas, whose parameters –e.g., the number of neighbors– have to be manually set, usually based on empirical evidences obtained in experiments. Model-based CF, in contrast, creates rating prediction

models whose parameters are fitted during a training phase, so that they minimize certain estimation error.

Among the existing model-based recommendation approaches, Matrix Factorization can be considered as the most successful and widely used.

Matrix Factorization (Koren et al., 2009) assumes that the user's preferences are determined by a number of unobserved (latent) factors. The items can also be described by this set of latent factors, and the more similar certain user and item latent vectors are, the higher the probability the user likes the item.

More specifically, let R be the $N \times M$ preference/rating matrix, whose element (i, j) corresponds to the rating $r(i, j)$ that user u_i has assigned to item i_j. The matrix R can be decomposed into two low-rank matrices U^t and V of size $N \times K$ and $K \times M$ respectively, with $K \ll M, N$, such that

$$R = U^t V$$

is the best approximate decomposition of R under a specific loss function. In this scenario, the user i is represented in a k-dimensional latent space at the i-th row of U^t matrix, $u_i \in \mathbb{R}^k$, and item corresponds to $v_j \in \mathbb{R}^k$. This k-dimensional space can be seen as a latent space where we can describe users and items over unobserved features. Once we have this representation, we can estimate ratings through the similarities between user and item latent factor vectors:

$$\hat{r} = u_i^t \cdot v_j$$

where the dot product will be higher as u_i and v_j are more similar.

This framework is very flexible, allowing the incorporation of aside effects or interactions. For example, we could add user and item rating biases, b_i and b_j, and a global average rating μ, so that the rating estimation becomes

$$\hat{r} = \mu + b_i + b_j + u_i^t \cdot v_j$$

which is the basic, standard model in Matrix Factorization for Collaborative Filtering.

To learn the factor vectors, u_i and v_j, the training algorithm minimizes a loss function, such as the regularized squared error on the set of training ratings:

$$\min_{u^*, v^*} \sum_{(u,i) \in \kappa} \left(r_{ui} - u_i^t v_j\right)^2$$

where κ is the set of pairs (u, i) whose rating r_{ui} is known. In order to avoid overfitting and allow the model to be able to generalize new unobserved ratings, it is appropriate to add a regularized term, so that the latent vectors do not have a high magnitude:

$$\min_{u^*,v^*} \sum_{(u,i)\in\kappa} \left(r_{ui} - u_i^t v_j\right)^2 + \lambda\left(\|u_i\|^2 + \|v_j\|^2\right)$$

where the hyper-parameter λ controls the degree of regularization, and is usually determined by cross-validation. In this case, the two main approaches (Koren et al., 2009) to minimize the loss are *stochastic gradient descent* (SGD) (Robbins and Monro, 1951) and *alternating least squares (ALS)* (Berge, 1993). SGD is easier and usually faster than ALS, that is more suitable when parallelization is available (Zhou et al., 2008) and with implicit data (Hu et al., 2008).

Similarly to content-based filtering, collaborative filtering has particular pros and cons. CF is able to provide diversity on the item recommendation lists, avoiding the *content overspecialization problem*. However, as CB, it also suffers from the *cold-start problem*; in this case, for both new users and new items. Differently to CB methods, in CF, an item is never going to be recommended unless some user has rated it. Another important disadvantage of CF if that they suffer from a *popularity bias*: the most popular items tend to be more recommended, and their popularity is further increased.

Hybrid recommender systems

As explained before, CB and CF approaches have specific, complementary problems. A straightforward solution to avoid some of such problems is by combining CB and CF methods. This is known as hybrid recommender systems.

There are several general ways of building hybrid recommenders (Adomavicius and Tuzhilin, 2005); some of them are:

- Executing CB and CF methods separately and combining their predictions, e.g. by means of as a weighted average.
- Incorporating content information as features in the CF latent factors vectors.
- Considering collaborative (rating-based) features in a content-based heuristic.

Hybrid recommenders have been proved to improve the performance over single types of recommendations, and thus most of the applications used in production consist of hybrid approaches, such as the well-known Netflix case (Gómez-Uribe and Hunt, 2015).

Other types of recommender systems

In addition to CB and CF systems, other types of recommender systems exist, such as:

- *Context-Aware Recommender Systems* (CARS), which consider not only the ratings, but also the contextual information when the items were rated, such as the current weather and time, and user's location, mood and social companion. The rationale of this type of recommenders is that users like different items in different contexts. For example, we may prefer a romantic and calm restaurant for a dinner

with our partner, and a cheap and crowded restaurant for having lunch with friends.

- *Knowledge-based Recommender Systems (KBRS)*, which use a set of pre-defined rules to generate recommendations based on inferences about the users' preferences. This type of recommendations is useful to overcome the cold start problem when there is a scarcity of user preference data.
- *Multi-criteria Recommender Systems (MCRS) (Adomavicius and Kwon, 2015)* considers the overall preference of a user over an item follows multiple criteria. In contrast with single-criterion value RS, where the utility function is defined over a single rating, MCRS takes a collection of ratings on different attributes of the item. For example, a user that rates a hotel with an overall score of 8 as a result of giving a 9 to *location* and a 7 to *cleanness* may not be considered as similar with a user that rates the hotel *cleanness* with 10 and its *location* with 6, even both users give the same overall rating to the hotel. In MCRS the user specifies in the query which are the criteria she is interested in and their corresponding value restrictions, for example, "get only items with location rating above 8". This additional information should improve recommendations since it captures more details on user's preferences.

2.2.2 Aspect-based recommender systems

Most of the methods that we have presented above only consider the overall rating of the items, ignoring the variety of opinions that users may have towards different aspects of the items. Aspect-based recommender systems (ABRS) consider these different opinions in order to improve item recommendations. There are multiple ways that aspect opinions may be used to predict overall ratings. For example, in the collaborative filtering approach, if a target user has agreed with others in the evaluation of particular aspects, these users' opinions on new items are very valuable to the target user. However, those opinions from reviewers that do not agree at aspect level (because they value different features) should not be considered as neighborhoods even assigning similar ratings to common preferred items.

Aspect-based RS may be view as an evolution of multi-criteria RS. In MCRS the user explicitly imposes some restrictions about attributes of the items, for example "restaurants with an average price below 25 and vegetarian food", very close to Information Filtering Systems. In ABRS users don't specify restrictions or filters but previous sentiment on features are considered to improve recommendations with this fine-grained information.

ABRS also share many elements with context-aware RS (CARS), since they consider not only the rating but additional information that can affect the recommendations. In CARS, recommendation models include context information, such as time of day,

weather or day of week. This information may affect the rating and the user could rate the item different in different contexts. The main difference with ABRS is that aspects are domain or even item-specific, whereas context is usually shared across the items. This small detail makes the frameworks to be quite different and CARS techniques are not usually used in ABRS.

2.2.3 Evaluation of recommender systems

At a theoretical level, an evaluation should replicate a real recommendation use case scenario, in order to choose the best recommender system for a particular goal. For example, Spotify may want the user to spend time listening to the recommended songs, instead of skipping them (Johnson, 2014), whereas Amazon's goal may be the user purchasing any of the suggested items (Linden et al., 2003). As there is a wide range of application use cases, the design of the experiments can also be very diverse.

Offline evaluation in Recommender Systems follows a similar methodology to classification, machine learning and information retrieval algorithms: use available data with information about the users' preferences as input for the algorithm (train data), and hold out a portion of such preferences (test data) to assess the quality of the recommendation model, in terms of how much the predictions resemble the true preferences in the test set.

Below we describe the most popular evaluation metrics in Recommender Systems and alternatives in the experimental setup.

Evaluation metrics

In the Recommender Systems research community, works have traditionally focused on measuring the accuracy of rating prediction, computing *error-based metrics*. However, authors found that this does not completely fits the real settings in working applications with deployed recommender systems, where the quality of a ranking of recommended items can be more effective than the accuracy of predicted rating values themselves (Bellogín, 2012). Hence, research community has moved from the annotation in context task (i.e., predicting ratings) to the find good items task (i.e., providing users with a ranked list of recommended items). As a result, *precision-oriented* metrics have being increasingly considered in the field.

Error-based metrics

In the context of explicit feedback, user preferences for items are represented as numerical ratings, and the goal of a recommendation algorithm consists of predicting unknown ratings. In this scenario, the accuracy of recommendations can be evaluated by measuring the difference between predicted and known ratings. The most popular metrics of this approach have been the **Mean Absolute Error** (MAE) (Shardanand and Maes, 1995) and the **Root Mean Squared Error** (RMSE) (Bennett et al., 2007):

$$MAE = \frac{1}{N}\sum_{i=1}^{n}(\hat{r}_i - r_i)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(\hat{r}_i - r_i)^2}$$

where $\hat{r}_i$ and r_i represents the predicted and real rating, respectively, and N is the dimension of test set. In general, RMSE is preferred to MAE since it penalizes larger errors.

The main limitation of these metrics is that errors in higher ratings penalize the same as errors in small ratings. In deployed RS, users are usually presented with items with a high predicted rating and it is desirable that errors at the top are considered differently. Besides, these metrics need the ratings to be numerical, so they can not be applied in implicit feedback recommender systems where user preferences are usually represented as a unary or binary ratings.

Precision-based metrics

Precision-based metrics in RS evaluation consider that users will be presented with a list of top-N ranked recommendations, which can be either relevant or not relevant to the user (Herlocker et al., 2004). If the rating scale is not binary, we need to transform it into a binary scale. For example, 1-5 ratings are usually transformed, such as 4 and 5 rating values are considered as "relevant", and 1-3 rating values as considered as "not-relevant." Considering Ret_u the list of returned items to user u, and Rel_u the set of relevant items to user u, some of the most popular metrics are (Bellogín, 2012):

- **Precision@k** *(Baeza-Yates and Ribeiro-Neto, 2011)* considers the percentage of k returned items that are relevant to the user.

$$P@k = \frac{1}{N}\sum_{i=1}^{n}\frac{|Rel_{u_i}|}{k}$$

- **Recall@k** *(Baeza-Yates and Ribeiro-Neto, 2011)* considers the percentage of the total relevant items to the user that have been returned:

$$R@k = \frac{1}{N}\sum_{i=1}^{n}\frac{|Rel_{u_i}@k|}{|Rel_{u_i}|}$$

- **Precision** and **Recall** are similar to their counterparts @k but, instead of using the cutoff, they consider the whole list of returned and relevant items, respectively.

- **Mean Average Precision** (MAP) (Manning et al., 2008) considers not only whether the returned items are relevant, but also their position in the ranking, by averaging $P@k$ for k being the position of every relevant document

$$MAP = \frac{1}{N}\sum_{i=1}^{n}\frac{1}{|Rel_{u_i}|}\sum_{j\in Rel_{u_i}} P@rank(u_i, j)$$

  where $rank(i,j)$ represents the position of item j in the ranking of user u_i.

- **Normalized discounted cumulative gain** (nDCG) (Järvelin and Kekäläinen, 2002) considers graded relevance that is being discounted if relevant items appear at lower positions at the ranking

$$nDCG@k = \frac{1}{N}\sum_{i=1}^{n}\frac{1}{IDCG_u^k}\sum_{p=1}^{k} f\big(rel(u_i, j_p), p\big)$$

  where $rel(u_{i,}, j_p)$ represents the relevance of item at position p for user u_i, and $f(x,y)$ is a discount function that grows with higher values of preference x and penalizes higher values of position y. $IDCG_u^k$ is the ideal $nDCG@k$ that corresponds to a perfect ranking sorted by descending preference.

- **User space coverage** (USC) measures the fraction of users for which at least k items will be returned. If the recommendation list does not require a minimum number of items, k is considered to be 1. In real applications, it may be convenient to have a tradeoff between coverage and quality of recommendations.

- **Item space coverage** (ISC) similarly measures the fraction of items for which the ratings of at least k users are predicted. This metric is related to the *diversity* of the recommender.

Other metrics from machine learning, such as Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, have also been used in some works (Herlocker et al., 2002), although they are much less frequent.

Experimental setups

Offline evaluation of RS follows the standard Machine Learning (ML) evaluation schema. The available data is split into *training set* and a *test set*, and frequently also a *validation set*. The training set is used to build the model, and the test set is used to feed the built model for computing metrics of performance. The validation set is used in the model building to fit its parameters.

Usually, ML observations are considered to be independently of each other, representing data from different individuals. However, recommender systems deal with multiple observations from the same user over the time. This is a very important difference that needs to be considered when building the test set, so the experiment is able to effectively reflect the available data the application would have in a real recommendation scenario.

Strategies that take time into account fix a point in time and split the dataset so that ratings before that time correspond to the training set, and the reminder ratings are considered as the test set. In this way, at the training phase, no future data is available, which is the case for real scenarios.

If we do not consider time, there are still some strategies to select the validation set, e.g., always selecting n items per user, selecting a variable (but fixed) number of items per user, or assigning a random sample that satisfies a global ratio on the whole dataset. There is not a consensus in the research community about which approach is preferred, but the last strategy seems to be the most popular (Bellogín, 2012).

Chapter 3

Related work

In this chapter we review published work related to the research problems addressed in this thesis, namely item aspect extraction and sentiment analysis, and aspect-based item recommendation. We describe existing approaches, discussing their advantages and disadvantages.

More specifically, in Section 3.1 we describe representative approaches to (item) aspect extraction from textual reviews, and in Section 3.2 we describe state-of-the-art aspect-based item recommender systems.

3.1 Aspect extraction and sentiment analysis

The automatic identification of references to aspects in textual reviews is a research problem related to Natural Language Processing. The consideration of grammatical patterns and syntactic sentence structures have been shown to be key issues for the effective extraction of aspects from text contents (Hu and Liu, 2004b).

Revising the research literature, we have identified two main types of approaches to aspect extraction. The first type is composed by those approaches that are based on *syntactic analysis* of the sentences in a review, while the second type is composed by those approaches that are based on *Topic Models*, which aim to group the words related to the same aspect. In the next subsections, we revise representative examples of such types of approaches.

3.1.1 Finding frequent nouns and compositional semantics

Considering word frequency

One of the simplest methods to extract references to aspects in textual reviews consists of directly identifying words that appear more often in a target domain than in a generic, multi-domain corpus. For instance, in a collection of reviews about restaurants, we shall find that words as 'ambience', 'service', 'food', 'dessert' or 'price' appear much more often than in document repositories on other domains.

In (Scaffidi et al., 2007) the authors build a Language Model to identify the aspects in reviews. They assume that item aspects are mentioned more often in a review than in generic English texts, so they compute the probability that word x is observed n_x times in a review of length N if the ratio of appearance in standard English is p_x. If the probability is high, then the word x is considered to be an aspect word. The opinion polarity is assigned based on the assumption that the global rating of a review correlates to the polarity of each word. Each aspect is then scored with the average polarity value assigned to the items it appears in.

A similar approach is followed in (Caputo et al., 2017), where the authors compare the words distributions in the target domain with their distributions in the British National Corpus[6]. In particular, they exploit the pointwise Kullback-Leibler divergence (KL-divergence), a non-symmetric measure of the difference between two points in two distributions. The words with a high score are considered to be the aspects.

Defining initial seeds

Another simple approach is to identify words related to previously defined aspect words. (Wang et al., 2010) and (Aciar et al., 2007) present semi-automatic methods that follows such methodology. In (Wang et al., 2010) the authors propose to start with a list of 'seeds' –i.e., a list of keywords– for each aspect, and iteratively enlarge it with words that appear in the reviews in the same context of the seeds and subsequently added keywords. In (Aciar et al., 2007) the authors build an initial ontology for each domain. The aspect, the keyword for that aspect, and a set of related words, compose the entities in the ontology. Sentences in the review are classified into positive, negative or about the author's experience with a classification model. The entities are then classified into the same categories according to the sentences they appear in.

These procedures are mostly automatic, but rely on initial seeds that need to be specified for each item type and domain.

---

[6] http://www.natcorp.ox.ac.uk

Using semantic dependencies

Other approaches use semantic dependencies to find the item aspects (Hu and Liu, 2004b; Popescu and Etzioni, 2005; Scaffidi et al., 2007; Qiu et al., 2011; Poria et al., 2014). They are based on the observation that aspects are usually nouns or noun expressions, and opinion words are mainly adjectives that act as modifiers of such nouns. This approach requires a preprocessing stage, including POS tagging, lemmatization, and constituent dependency. Most of them utilize the Stanford CoreNLP Parser[7] to do so.

In this context, (Hu and Liu, 2004b) is one of the first works on the extraction of item aspects from product reviews. The authors' goal is to summarize textual reviews, so readers could find the most useful ones, and highlight their most important parts. They employ association rule mining and the *Apriori algorithm* (Agrawal and Srikant, 1994) over nouns and noun phrases to find frequent *itemsets*, keeping the most frequent ones, and performing a pruning stage. The polarity of each aspect is assigned based on the adjectives that are closer to the found nouns. The aspects are annotated with the polarity (or its opposite) that the adjectives –or any of their synonyms or antonyms from WordNet – have in a lexicon composed of well-known opinion words[8].

The method presented in (Popescu and Etzioni, 2005) is based on the definition of an aspect as *part of* or *feature of* a product. It utilizes the system *KnowItAll Assessor* (Etzioni et al., 2005), which infers relationships such as *isPartOf(screen, phone)* by querying the Web. It computes the Point-Wise Mutual Information (PMI) between an entity f and a relation d as the ratio between the number of times (hits) they appear together with respect to the individual appearance,

$$PMI(f,d) = \frac{Hits(d+f)}{Hits(d) * Hits(f)}$$

In this context, the entities are the potential aspects to be extracted for an item, and the relations of type *isA()*. To assign the polarity to the aspect they first define a set of syntactic rules that extract opinion words from known aspect words. For example, the sentence *"I hate this scanner"* satisfies the rule *(subject, predicate, object)*. If we have already set *scanner* as an aspect, then *hate* is labeled as an opinion word. Then, once the potential opinion words have been identified, the semantic orientation and opinion is assigned to be the solution to a relaxation-labeling algorithm (Hummel and Zucker, 1983), with restrictions that come from the syntactic relations. For example, in the sentence *"iPhone 7 has a great camera and is very fast,"* the presence of the

[7] https://nlp.stanford.edu/software/lex-parser.shtml

[8] https://http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

conjunction 'and' implies that the words *great* and *fast* have the same polarity about the camera.

The *Double Propagation algorithm* (Qiu et al., 2011) is one of the most utilized state-of-the-art approaches to find aspect references in text reviews, solving the target aspect extraction and opinion expansion problems simultaneously. As continuation of previous works (Hu and Liu, 2004b; Popescu and Etzioni, 2005), the algorithm makes uses of the syntactic relations between nouns, or between noun sentences and adjectives. The Propagation Algorithms is described in Figure 3.1. It starts with a list of well-known opinion words, and their polarity, either positive or negative. Then it searches for words that are syntactically related to them according to the Universal Dependencies (UD) (Schuster and Manning, 2016) *mod*, *pnmod*, *subj*, *s*, *obj*, *obj2* and *des*. Those words are identified as *target words*, or words that are opinionated about.

This extraction is part of the first out of four rules that are meant to extract target and opinion words both from other target or opinion words:

- **Rule 1:** Extract a target word (noun) that is related to a known opinion word; or extract a target word related to another word that is also related to a known opinion word.
- **Rule 2:** Extract an opinion word (adjective) that is related to a known target word; or extract an opinion word related to another word that is also related to a known target word.
- **Rule 3:** Extract a target word that is related by a conjunction with a known target word; or extract a target word related to another word that has also a dependency of the same type with a known target word.
- **Rule 4:** Extract an opinion word that is related by a conjunction with a known opinion word; or extract an opinion word related to another word that has also a dependency of the same type with a known opinion word.

This propagation continues until neither more target nor opinion words are found. The obtained target words represent the potential extracted aspects, and a final pruning phase is conducted to remove the noise introduced in the propagation phase. In this phase, new aspect words are also added as a combination of two or more consecutive target words appearing together in the reviews.

The polarity of the aspects is computed at the propagation phase, and follows a general rule: new extracted words are assigned with the polarity of the known word used in the extraction, reversed if a negation term is present in the context of the known and extracted words.

There is an exception of this general rule of computing the polarity of a word. If an opinion word is extracted from a target word that appears in a different review, we

cannot assume that author's opinion is maintained in this review and we assign the opinion word the average review polarity instead of target polarity. The average polarity in a review is computed as the average of opinion words' polarities that are contained in it. This modification leads to the possibility that an opinion or target word gets several polarities. The final polarity is computed from the average.

Double Propagation is very extended in aspect-based recommender systems, such as (Bauman et al., 2016; Wang et al., 2012). In (Poria et al., 2014), the authors present an improved version of the rules used in the propagation stage, and also extract "implicit" aspects, being the first to do it, to the best of our knowledge.

Input: Opinion Word Dictionary $\{O\}$, Review Data R
Output: All Possible Features $\{F\}$, The Expanded Opinion Lexicon $\{O\text{-}Expanded\}$
Function:
1. $\{O\text{-}Expanded\} = \{O\}$
2. $\{F_i\} = \emptyset, \{O_i\} = \emptyset$
3. for each parsed sentence in R
4. if(Extracted features not in $\{F\}$)
5. Extract features $\{F_i\}$ using $R1_1$ and $R1_2$ based on opinion words in $\{O\text{-}Expanded\}$
6. endif
7. if(Extracted opinion words not in $\{O\text{-}Expanded\}$)
8 Extract new opinion words $\{O_i\}$ using $R4_1$ and $R4_2$ based on opinion words in $\{O\text{-}Expanded\}$
9. endif
10. endfor
11. Set $\{F\} = \{F\} + \{F_i\}$, $\{O\text{-}Expanded\} = \{O\text{-}Expanded\} + \{O_i\}$
12. for each parsed sentence in R
13. if(Extracted features not in $\{F\}$)
14. Extract features $\{F'\}$ using $R3_1$ and $R3_2$ based on features in $\{F_i\}$
15. endif
16. if(Extracted opinion words not in $\{O\text{-}Expanded\}$)
17. Extract opinion words $\{O'\}$ using $R2_1$ and $R2_2$ based on features in $\{F_i\}$
18. endif
19. end for
20. Set $\{F_i\} = \{F_i\} + \{F'\}$, $\{O_i\} = \{O_i\} + \{O'\}$
21. Set $\{F\} = \{F\} + \{F'\}$, $\{O\text{-}Expanded\} = \{O\text{-}Expanded\} + \{O'\}$
22. Repeat 2 till size($\{F_i\}$) = 0, size($\{O_i\}$) = 0

Figure 3.1 Propagation algorithm from (Qiu et al., 2011)

3.1.2 Using topic models

The output of the analysis approach described in the previous Section is a list of (mostly) nouns and noun phrases that represent aspects of reviewed items. As analyzed in Section 0, one of the main drawbacks of such approach is that the extracted aspects words are considered to be independent from each other, even though some of them may be related. For example, users may talk about the 'service' in a restaurant by using distinct words like *service*, *staff* and *attention*, which should not be considered as different aspects. To overcome this issue, clustering methods are a possible solution, so that related words are grouped together and assigned to the same aspect.

Hence, several strategies have been proposed to group together related words after running a semantic aspect extraction process; most of them based on Topic Models algorithms.

An example of this approach is (Wang et al., 2012), where reviews are first annotated by the Double Propagation algorithm (explained in Section 0), and then the annotated reviews are used as input of the LDA technique, which clusters the aspects terms into latent factors (assumed as aspects). In this case, the score associated to each aspect cluster is computed as the ratio of number of positive words with respect to the total number of opinion words about the aspect.

Differently to (Wang et al., 2012), the majority of other proposed methods rely on topic models for both extracting and clustering aspect-related words in a single phase (McAuley et al., 2012; McAuley and Leskovec, 2013; Titov and McDonald, 2008a, 2008b; Wang and Blei, 2011; Wu and Ester, 2015; Zhao et al., 2010). When applied directly, these methods are not able to capture the appropriate item aspects. In particular, they tend to build general topics that classify terms into instances of the items the reviews talk about. For example, in the restaurants domain, topics are usually related to types of cuisine, such as Italian, Asiatic, vegetarian and vegan; in movies and books reviews, topics in general correspond to genres; and in electronics reviews, topics tend to represent different types of devices.

To overcome the above issue, some authors have developed adapted versions of the generative process in LDA, somehow guiding the generation of useful topics.

In (Titov and McDonald, 2008a), the authors present the Multi-Grain Topic Models (MG-LDA) algorithm, where a word is generated as a sample of either a mixture of Global Topics or a mixture of Local Topics depending on the word context. The MG-LDA model is shown in Figure 3.2 and the formal generation process is as follows:

First, for each document d:

- Choose a distribution of global topics that appear in the document $\theta_d^{gl} \sim Dir(\alpha^{gl})$
- For each sentence s, choose a distribution $\psi_{d,s}(v) \sim Dir(\gamma)$
- For each sliding window v:
    - Choose a distribution of local topics $\theta_{d,v}^{loc} \sim Dir(\alpha^{loc})$
    - choose $\pi_{d,v} \sim Beta(\alpha^{mix})$, the prior distribution for choosing between local and global topics

- For each word i in sentence s of document d:
    - choose window $v_{d,i} \sim \psi_{d,s}$
    - choose $r_{d,i} \sim \pi_{d,v_{d,i}}$ representing whether a word becomes from the global or local distribution
        - if $r_{d,i} = gl$ choose global topic $z_{d,i} \sim \theta_d^{gl}$
        - if $r_{d,i} = loc$ choose local topic $z_{d,i} \sim \theta_{d,v_{d,i}}^{loc}$
    - choose word $w_{d,i}$ from the word distribution $\varphi_{z_{d,i}}^{r_{d,i}}$ of local or global topics that comes from a Dirichlet prior $Dir(\beta^{loc})$ or $Dir(\beta^{gl})$.

This approach improves the quality of the LDA by considering as aspects only those topics that can be explicitly rated, and excluding those topics that are generic. The ratable aspects are captured by local topics, and global topics are related to general properties of reviewed items. For example, in reviews about hotels in London, a global topic that will emerge is London itself, but that is an aspect that will not be evaluated. MG-LDA is not addressed to estimate the rating of this aspect and need to be done externally. Authors compute the predicted score for each aspect in a supervised fashion. They compute a set of features for each review, including top 3 and run PRanking algorithm (Crammer and Singer, 2001), a multi-aspect rater that runs a perceptron-based classifier, for each aspect, trying to recover the assigned numeric rating.

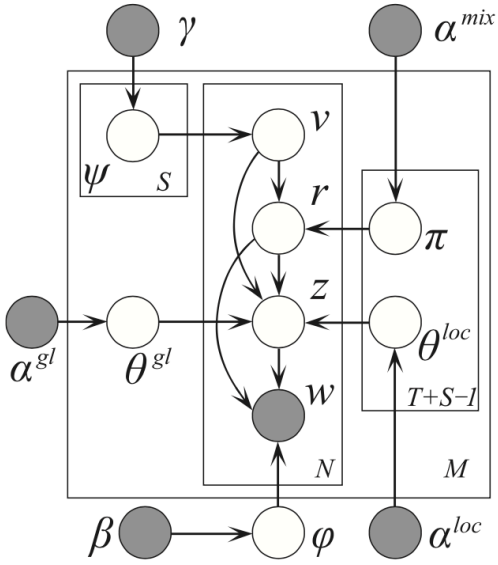

Figure 3.2 MG-LDA model from (Titov and McDonald, 2008a), where LDA model is extended to consider a mix of global $\boldsymbol{\theta^{gl}}$ and local $\boldsymbol{\theta^{loc}}$ topics

The same authors propose in (Titov and McDonald, 2008b) a method that is able to associate the topics obtained with MG-LDA with a particular item aspect. That is one of the main challenges of LDA-like methods: there is not a one-to-one correspondence between topics and aspects, and a topic may refer to several aspects. The procedure is based on the assumption that aspect ratings should be correlated with item ratings. Hence, the global rating of the review may be helpful to separate topics that correspond to different aspects. It runs in a single phase where the topics learnt by MG-LDA also depend on the review ratings. First, each aspect is associated with a global rating based on the review ratings where it appears, and then it is modified based on the specific words that comment on the aspect. The results show, for an experiment on hotel reviews, that the first three local topics found by the algorithm correspond one-to-one to the aspects *service*, *location* and *rooms*, as desired.

That is the first work that shows that modeling aspects and ratings at the same time improves the quality of the aspects found, and represents the beginning of a research line that is very active as of today.

In this line, Zhao et al. (2010) present a modified version of MG-LDA that considers three types of words, namely *aspect words*, *opinion words*, and *background words*. These options are considered in the generative process of a word. The probability of a word being of any of those three classes does not follow a symmetric Dirichlet prior, since it is very related to its syntactic category. Hence, the probability of a word being *aspect*, *opinion* or *background* is estimated fitting a maximum entropy (MaxEnt) model from $\boldsymbol{x}_{d,s,n}$, a feature vector for n-th word $w_{d,s,n}$ of sentence s of document d. The parameters are learnt from a set of training sentences labeled with background, aspect and opinion words. $x_{d,s,n}$ can encode arbitrary features that may be discriminative of

the word type. Authors use lexical features (previous, current and next words) and POS tag features (previous, current and next POS tags). They prove with the experiments that POS tag features are very predictive of the word type.

In (McAuley et al., 2012) the authors present an unsupervised method that separately models words that discuss an aspect from words that discuss the associated sentiment, by generating a word from one of both distribution. They fit the parameters from an annotated corpus that contains ratings at the aspect level.

As a step forward in this line, Topic Modeling and Matrix Factorization (MF) for Collaborative Filtering (CF) have been combined in order to use the topic distribution as part of the latent factor vectors for user and items. We provide more details about this approach in Section 3.2, but here we highlight the main characteristics of some related works.

This procedure has been used in (Wang and Blei, 2011) to recommend scientific articles to researchers through a Collaborative Topic Regression (CTR) model. As in standard Matrix Factorization for Collaborative Filtering, the user is represented by means of latent factors. In this scenario, each factor is associated to a particular aspect. The key modification of MF is at the item level, where an item latent factor is constrained to be close to the topic proportions derived from the item review text. It can be interpreted as a combination of content based –represented by the topic proportions– and collaborative filtering data.

A slightly modified version of the CTR model is presented in (McAuley and Leskovec, 2013). Instead of including review topics and rating information in the same model, the authors fit two separated models for the ratings and the topics. Afterwards, the item latent vector and the topic distribution for the item are forced to be tight together.

Finally, a 5-component model is proposed in (Diao et al., 2014) for the word distribution generative process. Words in the reviews may come from a background language model, a background sentiment distribution, an item-specific word distribution, an aspect-specific word distribution, or an aspect-specific sentiment distribution. The standard user-item latent model is modified such as it includes the agreement between the aspects the user cares about and the item aspects

3.2 Aspect-based recommendation

Aspect-based recommender systems aim to provide personalized recommendations taking into account the users' opinions about aspects of the rated items. For example, let us consider a user who is concerned about the audio characteristics of electronic devices. When such user receives recommendations about mobile phones, she may find valuable only those corresponding to phones that have good voice quality. In this

context, a recommender system that considers mobile phone aspects could be able to capture such particular preference, and suggest the user with devices that satisfy her.

The work (Aciar et al., 2007) is, to the best of our knowledge, the first attempt to estimate the rating for each item aspect, and provide recommendations based on the aspects ratings. Analyzing reviews, the authors first obtain the sentences related to each aspect, and then heuristically define a way to compute the aspect ratings based on the sentence positivity or negativity, and the reviewers' reputation.

Further works considering item aspect in the recommendation process are based Collaborative Filtering, especially using the Matrix Factorization technique, e.g. (Wang and Blei, 2011; Wang et al., 2012; McAuley and Leskovec, 2013; Bauman et al., 2016).

3.2.1 Global rating as a combination of aspect-specific ratings

Some works have considered the global rating that a user assigns to an item to be a weighted combination of the ratings that she would assign to the item aspects. For example, Wang et al. (2012) extract the aspects from a review through a combination of Double Propagation and LDA. Then, they build a matrix for each aspect, where each entry is computed as the ratio of the number of positive words over the total number of opinion words about the aspect. These matrices are integrated into a K-dimensional tensor (where K is the number of found aspects), and a Tensor Factorization model is used to make the recommendations.

In (Wang et al., 2010) the authors model the global rating by means of a Bayesian Regression on the observed aspects ratings. The ratings are considered to follow a normal distribution whose mean is a weighted combination of the ratings of the aspects. These ratings are generated as another weighted combination of the words in the reviews with opinions on the aspect. A very similar approach is followed by (Wu and Ester, 2015).

Finally, Bauman et al. (2016) fit two different models to estimate global item ratings. Previously, they extract the sentiment of each aspect s_{ij}^{k} by following the Double Propagation algorithm, and compute the sentiment on each unrated aspect t as a weighted sum of the sentiment of the k nearest neighbor aspects. The weights w_{tj} are compute as the Spearman's correlation score between the two aspects.

$$\hat{s}_{ij}^{t} = \frac{\sum_{1}^{k} w_{tk'} \cdot s_{ij}^{k'}}{\sum_{1}^{k} w_{tk'}}$$

Then, they fit a standard Matrix Factorization model for each aspect,

$$\hat{s}_{ij}^{t} = \mu^{t} + b_{i}^{t} + c_{j}^{t} + u_{u}^{t} \cdot v_{j}^{t}$$

and build a regression model to estimate the global rating from the estimated sentiment for each aspect

$$r_{ij} = \left(A + B_i + C_j\right) \cdot S_{ij}$$

3.2.2 Aspect-based user and item representations

A simple, yet effective strategy is to exploit extracted aspect information as representations of user preferences and item attributes, which are incorporated into traditional Collaborative Filtering heuristics. Specifically, similarity metrics between users (items) are computed on top of such representations. This is the approach followed in (Musto et al., 2017), where an item is represented by all its aspects mentioned in the reviews, and a user is represented by means of the aspects she has commented on some review.

Recall the rating estimation equation for user-based CF presented in Section 0.

$$\hat{r}(u,i) = c \sum_{u_k \in \eta_n(u)} r(u_k,i) \cdot sim(u,u_k)$$

where users u and u_k are represented as the vector of ratings they assigned to each item $\left(r(u,i_1),\dots,r(u,i_n)\right)$. In aspect-based approaches, the user representation contains information about the item aspects, and the opinion the user has expressed about them.

Musto et al. express the similarity of two users as the opposite of their distance

$$d\left(u_j,u_k\right) = \frac{1}{|I\left(u_j,u_k\right)|} + \sum_{i \in I\left(u_j,u_k\right)} d\left(R\left(u_j,i\right),R(u_k,i)\right)$$

where $I\left(u_j,u_k\right)$ is the set of the co-rated items and $R(u_n,i)$ is the relevance scores of user u_n on each aspect of item i that are obtained from computed KL divergences. The overall distance between two aspect-relevance representations that share n aspects a is calculated as the L2 distance

$$d\left(R\left(u_j,i\right),R(u_k,i)\right) = \sqrt{\sum_{a=1}^{n} \left|R_a\left(u_j,i\right) - R_a(u_k,i)\right|^2}$$

The authors follow a similar approach for item-based CF using aspect-based representations of the items.

The final estimated ratings $\hat{r}(u,i)$ are predicted using the standard weighted sum considering the global rating.

3.2.3 Topic Models for item latent factors

As introduced in Section 3.1, another way to exploit item aspect sentiment information for recommendation is to consider the topic distribution vectors extracted from the reviews *into* collaborative filtering based on latent factor models. This combined method improves the recommendations with respect to using only CF, and is the state-of-the-art approach for aspect-based item recommendations based on topic models. The

topic model approach provides a content-based component that helps, for example, to consider items with very few reviews (i.e., the cold start), since there is much more information extracted from few reviews than considering only their ratings (Ganu et al., 2009)

In (Wang and Blei, 2011) the authors modify the item latent vector in standard MF to be $v_j = \epsilon_j + \theta_j$ where θ_j is the topic proportion vector, and ϵ_j is the standard item latent offset. This method is called Collaborative Topic Regression (CTR), and introduced a new paradigm of aspect-based recommender systems.

The standard MF algorithm explained in 0 can be interpreted as a Probabilistic Matrix Factorization (PMF) with the following generation process:

1. For each user u_i, draw a user latent vector $u_i \sim N(0, \lambda_u^{-1} I_K)$.
2. For each item v_j, draw an item latent vector $v_j \sim N(0, \lambda_v^{-1} I_K)$.
3. For each user-item pair (u_i, v_j), draw the response $r_{ij} \sim N(u_i^t v_j, c_{ij}^{-1})$. Where c_{ij} is a parameter that represents the confidence about r_{ij}.

The graphical model for CTR is shown in Figure 3.3. Assuming there are K topics, $\beta_{1:K}$, the generative process is an adapted version of PMF that includes the topic distribution, as follows:

1. For each user i, draw user latent vector $u_i \sim N(0, \lambda_u^{-1} I_K)$.
2. For each item j,
    a. Draw topic proportions $\theta_j \sim Dir(\alpha)$
    b. Draw item latent offset $\epsilon_j \sim N(0, \lambda_v^{-1} I_K)$ and set the item latent vector as $v_j = \epsilon_j + \theta_j$.
    c. For each word w_{jn},
        i. Draw topic assignment $z_{jn} \sim Mult(\theta)$.
        ii. Draw word $w_{jn} \sim Mult\left(\beta_{z_{jn}}\right)$.
3. For reach user-item pair $\left(u_i, v_j\right)$, draw the rating $r_{ij} \sim N(u_i^T v_j, c_{ij}^{-1})$

Note that the key part is the generation of the item latent vector v_j (step 2.b.), which differs from MF and is forced to be close to topic proportions θ_j, but could diverge from it if it has to. The parameters of the model are estimated with MAP through an EM-style algorithm.

The authors evaluated this model in a corpus of scientific articles. The obtained results show that CTR is slightly better than traditional Collaborative Filtering. CTR outperforms LDA in out-of-matrix prediction, a task where MF is unable to work. They also show that adding content into CTR improves performance over MF for in-matrix

predictions as well. In particular, they show that greater improvements are achieved when the number of returned items is larger. An explanation for this phenomenon is that CF works well for popular items, but when a larger number of items is required, there are few user ratings to ensure the quality of CF recommendations, and the content contribution becomes more important.

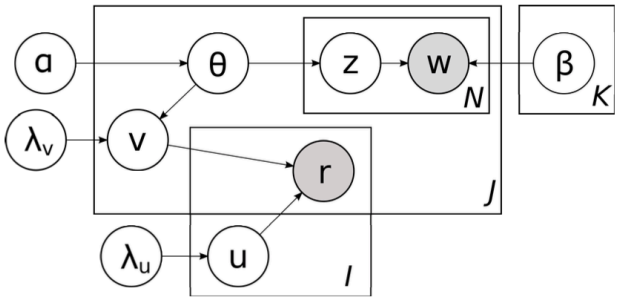

Figure 3.3 The graphical model for the CTR model

In (McAuley and Leskovec, 2013) the authors aim to understand and explain the hidden dimensions of latent factors models for the ratings. They assume that there is a relation between the latent factors and the latent dimensions of topic models that associates each topic with each factor. They present the Hidden Factors as Topic (HFT) model that aims to discover topics that are correlated to the hidden factors of products and users. The idea is to fit the LDA and the MF models in parallel, and tight the topic distribution and latent vector together during the learning phase. This is done by a linear transformation between the k component of the topic proportions θ_j and the parameters of the MF item vector v_j by a monotonic transformation,

$$\theta_{j,k} = \frac{\exp(\kappa v_{j,k})}{\sum_{k'} \exp(\kappa v_{j,k'})}$$

where the parameter κ controls the peakiness of the transformation: higher values implies θ approximates the largest component of v, and smaller values lead to a uniform distribution. In the experiments, they show that this model improves the latent factor model when the available data is small.

Diao et al. present in (Diao et al., 2014) a variation of Probabilistic Matrix Factorization (PMF). The global rating is modeled as

$$\hat{r}_{ij} = u_i^T \left[\sum_a p(a|\theta_i, \theta_j) M_a \right] v_j + b_0 + b_i + b_j$$

where b_0, b_i and b_u are the global, user and item biases, respectively; u_i and v_j are the user and item latent vectors, M_a captures aspect-specific properties, and the expression in the summation is the probability that the aspect-based information about the user and the item do agree on aspect a.

Chapter 4

Developed methods

In this chapter we present the methods developed in this thesis for the aspect extraction and aspect-based recommendation tasks. Evaluating these methods we will aim to understand how existing aspect extraction approaches behave on several domains, and how the exploitation of aspect sentiment information affects the performance of several recommendation techniques. Moreover, we propose a new aspect-based recommendation algorithm based on a state-of-the-art matrix factorization model for collaborative filtering.

More specifically, in Section 4.1 we first describe the developed methods for identifying aspects in textual reviews, and their corresponding opinion polarities. Next, in Section 4.2, we present the evaluated recommendation methods, including those we propose to exploit aspect sentiment information.

4.1 Aspect extraction methods

In this Section we explain the developed methods to identify user opinions about item aspects from textual reviews. In particular, to compare the main approaches described in Section 3.1, we have implemented a method based on semantics relationships and other method based on topic models. For evaluation purposes, we have also implemented a baseline method that uses manually defined initial seed words that describe a number of fixed item aspects.

4.1.1 Aspect extraction based on initial seeds

The simplest approach for aspect-based recommendation is to make use of a small, fixed and manually selected set of aspects. As a baseline aspect extraction method we follow this idea.

For a target domain, we first checked popular websites where items are exhaustively described, analyzed or reviewed by means of particular attributes and evaluation criteria to set an initial set of 'seed' words referring to item aspects. We also performed a manual inspection of available reviews in such domain in order to validate and extend the generated list of seeds.

As we shall show in Chapter 5, in our experiments we used a dataset of Amazon reviews about products belonging to five domains, namely books, movies, music, cell phones, and video games.. Table 4.1 shows the manually selected sets of aspects for each of such domains.

Domain	**Number of aspects**	**Aspects**
Books	10	characters, coherence, descriptions, ending, literary style, pacing, pictures, price, script, story
Movies	16	art style, cast, characters, costumes, direction, ending, locations, music, pacing, photography, picture, price, script, sounds, story, visual effects
Music	10	dynamics, harmony, lyrics, melody, price, rhythm, sounds, style, texture, timbre
Cell phones	16	appearance, battery, buttons, camera, charger, connectivity, memory, microphone, price, processor, protector, screen, size, sound, speaker, weight
Video games	15	art style, characters, controls, customization, difficulty, gameplay, ending, graphics, music, pacing, price, script, sounds, story, visual effects

Table 4.1 Sets of aspects manually selected for each domain.

In the table, each aspect is represented by a single word. However, in practice, an aspect was represented by several words. For instance, the 'sounds' aspect in the music domain was referred by words such as 'sounds', 'sound effects', 'audio effects', 'digital sounds', 'voices' and 'audio'. Moreover, for a particular word, we also considered morphological deviations, such as 'sound effect', 'sound effects', 'sound-effect' and 'sound-effects' for the 'sound effects' seed word.

After the initial sets of words representing aspects were built, we performed an automatic process to extend them with synonyms. Specifically, for each seed word, we retrieved its synonyms in the WordNet dictionary (Miller, 1995). In order to avoid noise, we considered the synonyms of a limited number of meanings (*synsets* in WordNet). In particular, we just took into consideration those meanings for which the definition of the target word contained certain domain-dependent words, such as 'music' and 'musical' for the music domain. Thus, for instance, to retrieve synonyms for the word 'tone', we considered the synonyms of the following synsets:

- "(music) the distinctive property of a complex sound (a voice or noise or musical sound)"
- "a notation representing the pitch and duration of a musical sound"

and not others such as:

- "the general atmosphere of a place or situation and the effect that it has on people
- a quality of a given color that differs slightly from another color."

4.1.2 Aspect extraction based on semantic relationships

We have implemented the Double Propagation method (Qiu et al., 2011) described in Section 0. The method uses the Opinion Lexicon from (Hu and Liu, 2004b) as the input repository of opinion words, with positive and negative polarities. The method uses a set of rules to expand the lists of (potential) aspects and opinion words from previously identified aspect words.

We have used the Stanford CoreNLP[9] library for both POS tagging and constituent and dependency parsing. Contrarily to (Qiu et al., 2011), we also have used the CoreNLP framework for sentence parsing and sentence clause extraction. More specifically, to obtain the clauses of a sentence, we have executed the Constituency Parser to obtain the Treebank of a sentence, and from it we have retrieved the subtrees whose roots start with 'S' (S/SBAR) and do not contain a subtree starting with 'S', meaning they are simple declarative clauses.

The **propagation algorithm** was shown in Figure 3.1. It first extracts new features (aspects) and opinion words based on known opinion words, and afterwards based on extracted features. In the implementation of the algorithm, we consider two words to be the same if they share the same lemma.

[9] https://stanfordnlp.github.io/CoreNLP/, https://nlp.stanford.edu/software/lex-parser.shtml

The **opinion polarity** assigned to a new extracted feature follows a general basic rule: it acquires the Lexicon polarity value of the corresponding word. However, we reverse the polarity value if a negation is present in the context of the word, i.e., if the associated adjective is negated (e.g., non-melancholic lyrics), or the sentence or phrase of the clause is negative (e.g., has not melancholic lyrics). We have also considered exceptions and particular cases specified in Qiu et al.'s work.

The algorithm has a subsequent **pruning phase** that aims to remove those nouns that have been detected as aspects, but are mostly noise. We have implemented two variations of this pruning: one at clause level and other at sentence level. They consist on keeping the most frequent words (in the domain) in case more than one noun in a clause are identified as aspects. As part of the pruning phase, we have also identified *target phrases* by combining each target word with up-to-Q consecutive words right before and after the target word, and K adjectives before the target word. We set Q=2, K=1 as in (Qiu et al., 2011). Finally, we have run a *global pruning* where every identified target that appears only once in the corpus is removed. We do not run the *product pruning* proposed by Qui et al. since they explained that it trades recall for precision without an improvement of the F-score, and the other pruning stages are already quite strict.

4.1.3 Aspect extraction based on topic models

We also want to test how topic models for aspect extraction behave. We run LDA on each domain. We consider the set of all reviews of a particular item as a document, as suggested in (McAuley and Leskovec, 2013). We use the LDA implementation in MALLET[10] framework. We run LDA for 5, 10, 20 and 50 topics to analyze this effect.

The output of the topic model procedure is a word distribution for each topic and a topic distribution for each item (document). The polarity assigned to each aspect (topic) is the average polarity of the closest opinion word to each word defining the aspect, weighted by the relevance of the word into the topic.

4.2 Aspect-based recommendation methods

In this Section we describe the evaluated aspect-based recommendation methods, and propose a matrix factorization collaborative filtering model that exploits item aspect-based user preferences, and a hybrid recommendation method that uses representations of item aspects in the similarity between users and items.

[10] http://mallet.cs.umass.edu

We have developed and evaluated **ATagMF**, a new aspect-based matrix factorization model based on the TagGSVD++ model presented in (Fernández-Tobías, 2017) for cross-domain recommendation.

The TagGSVD++ model aims to recommend items in a target domain by using information from another source domain. For instance, a user who has liked 'romance' movies in the past it is very likely to like 'romantic' and 'melancholic' music as well. Previous work (e.g. Golder and Huberman, 2006) had shown that social tags assigned by users to items in websites represent both user preferences and item features, so they could be used to enhance personalized recommendations. Upon this observation, Fernández-Tobías and others have investigated the exploitation of social tags in recommendation in general, and to transfer knowledge between domains in particular.

TagGSVD++ is an extension of ItemRelTag model in (Enrich et al., 2013), which considers the set of relevant tags assigned by the whole community to a target item. More specifically, it also considers user preferences expressed in the tags assigned by the target user to other items. For such purpose, Fernández-Tobías adapts the gSVD++ algorithm (Manzato, 2013), which considers items attributes in addition to the user's feedback, by including an additional set of latent variables. Hence, the estimation of the rating from user i to item j is as follows:

$$\hat{r}(i,j) = \left\langle u_i + \frac{1}{|T_i|}\sum_{s\in T_i} n_{is}\,\vec{x}_s\,,\; v_j + \frac{1}{|T_j|}\sum_{t\in T_j} n_{jt}\,\vec{y}_t \right\rangle$$

where u_i and v_j are the user and item latent vectors of the Matrix Factorization model, respectively. The tags are represented in the same latent space than users and items in MF, by means of the latent vectors $\vec{x}_s$ and $\vec{y}_t$. T_i and T_j are the set of tags assigned by user i to every item, and to item j by the whole community of users; n_{is} is the number of items on which user i applied tag s and n_{jt} the number of users that applied tag t to item j. Tag factors are normalized by $|T_i| = \sum_{s\in T_i} n_{is}$ and $|j| = \sum_{t\in T_j} n_{jt}$ so that factors $\vec{x}_s$ and $\vec{y}_t$ do not dominate over the rating factors u_i and v_j for users and items with a large number of tags.

The parameters are learned from the observed data by minimizing the regularized squared loss function

$$\mathcal{L}\,(\boldsymbol{U},\boldsymbol{V},\boldsymbol{X},\boldsymbol{Y}) = \sum_{(i,j)\in\mathcal{R}} \frac{1}{2}\left(r_{ij} - \left\langle u_i + \frac{1}{|T_i|}\sum_{s\in T_i} n_{is}\,\vec{x}_s\,,\; v_j + \frac{1}{|T_j|}\sum_{t\in T_j} n_{jt}\,\vec{y}_t \right\rangle \right)^2 + \frac{\lambda}{2}\left(\|u_i\|^2 + \|v_j\|^2 + \sum_{s\in T_i}\|\vec{x}_s\|^2 + \sum_{t\in T_j}\|\vec{y}_t\|^2 \right)$$

Stochastic Gradient Descent is used to find a local minimum of $\mathcal{L}$ by iteratively updating the parameters after each observed $(u, i) \in \mathcal{R}$ pair. More details on the update equations can be found in (Fernández-Tobías, 2017).

We propose to adapt TagGSVD++ for aspect-based recommendation. In particular, the set of tags is replaced by set of aspects that have been automatically extracted from textual reviews. Differently to social tags, which are in general associated with positive user preferences, aspects may also have a negative polarity. As a proof of concept, in this work we will only use the aspects that have been identified with positive sentiments in the reviews. We leave for future work the consideration of negative aspect-based user preferences.

Apart from the *ATagMF* model, we also evaluate content-based approaches based on the aspect extracted through the methods proposed in Section 4.1.

First, we evaluate **DP-CBCF**, a hybrid content-based collaborative-filtering recommender system that uses the extracted aspects through Double Propagation. Items are represented with the set of total aspects found in a domain, and the entries are the average of the estimated polarity over every time the aspect appears in the reviews. User vectors are computed similarly by aggregating opinions the user has on each aspect she comments on. We also evaluate a pure content-based approach where users and items are described as above.

Finally, we also test a content-based and hybrid CB-CF using the aspect representations obtained through LDA, **LDA-CBCF**. In this case, the items are represented by the topic distribution. The user profiles are computed as the weighted average over the items they have rated, considering the rating as the weight.

We have implemented these methods on top of the RankSys framework[11].

[11] http://ranksys.org

Chapter 5

Experiments

In this Section we report the experiments conducted to evaluate the developed methods presented in Section 4. In Section 5.1 we describe the used datasets, and in Section 5.2 we explain the followed evaluation methodology and used metrics. Finally, in Section 5.4 we discuss the obtained empirical results.

5.1 Datasets

We have evaluated the developed methods on two collections of datasets, both containing Amazon user reviews about products in several domains.

We have utilized a first dataset collection to validate the implementation of the Double Propagation aspect extraction method (Hu and Liu, 2004b). This dataset was used in (Hu and Liu, 2004b) to evaluate the above method, and contains a small number of reviews from five particular products: two digital cameras, a DVD player, an audio player, and a mobile phone. We will refer to these products as 'Canon', 'Nikon', 'Apex', Jukebox' and 'Nokia', respectively. In (Hu and Liu, 2004b) the reviews in the dataset were manually annotated with the aspects present in each sentence, and their opinion polarities. Descriptive statistics about these datasets are shown in Table 5.1. All of them contain less than 100 reviews, with a total number of sentences ranging from 350 to more than 1750 approximately. There are around 100 real aspects per item.

	Apex	**Canon**	**Jukebox**	**Nikon**	**Nokia**
Number of reviews	99	45	95	34	41
Number of sentences	726	595	1747	355	557
Number of true aspects	**110**	**99**	**179**	**74**	**107**

Table 5.1 Statistics of the annotated reviews datasets.

More specifically, we have used these small, manually annotated datasets to test the effect of pruning on the Double Propagation algorithm, and check how this algorithm and LDA behave.

The second collection of datasets contains several much larger public available sets of Amazon reviews[12] about products belonging to different domains, which do not have manual annotations of aspect opinions. It is an improved version of the data compiled in (McAuley and Leskovec, 2013). In particular, we have selected the reviews about "CDs and Vinyl", "Digital music", "Movies and TV", "Phones" and "Videogames" *5-core* categories. An example of a user review about a videogame is shown in Figure 5.1. It contains the *ids* of the reviewer and the item (product), the reviewer's *nickname*, a flag indicating whether the review had been marked as *helpful*, the review *text*, the *rating* assigned by the reviewer to the item, a *summary* of the review acting as its title, and the review *time* in UNIX string format.


```
{
    "reviewerID": "A22KRTIWDLOA98",
    "asin": "7100027950",
    "reviewerName": "chadwick",
    "helpful": [0,0],
    "reviewText": "Great game! I love the storyline and graphics, as well as the fighting
style. Minus the super long time it takes traveling the ocean, this game is a blast, and a
must-have for any fan of the Zelda franchise.",
    "overall": 5.0,
    "summary": "Epic Zelda title!",
    "unixReviewTime": 1333497600,
    "reviewTime": "04 4, 2012"
}
```


Figure 5.1 An example of review about a videogame.

Descriptive statistics about these datasets are shown in Table 5.2. We can observe that there are significant differences between the datasets, with total numbers of reviews in each domain ranging from around 65 thousands to almost 9 million. The rating distribution is shown in Figure 5.2; more than 50% of the ratings correspond to 5 stars, a well-known positive bias that usually occurs in rating-based user feedback.

[12] http://jmcauley.ucsd.edu/data/amazon

	Books	CDs and Vinyl	Digital music	Phones	Videogames
Number of reviews	8,898,0410	1,097,592	64,706	194,439	231,780
Number of items	367,982	64,443	3,568	10,429	10,672
Number of users	603,668	75,258	5,541	27,879	24,303

Table 5.2 Statistics on 5-core Amazon reviews datasets.

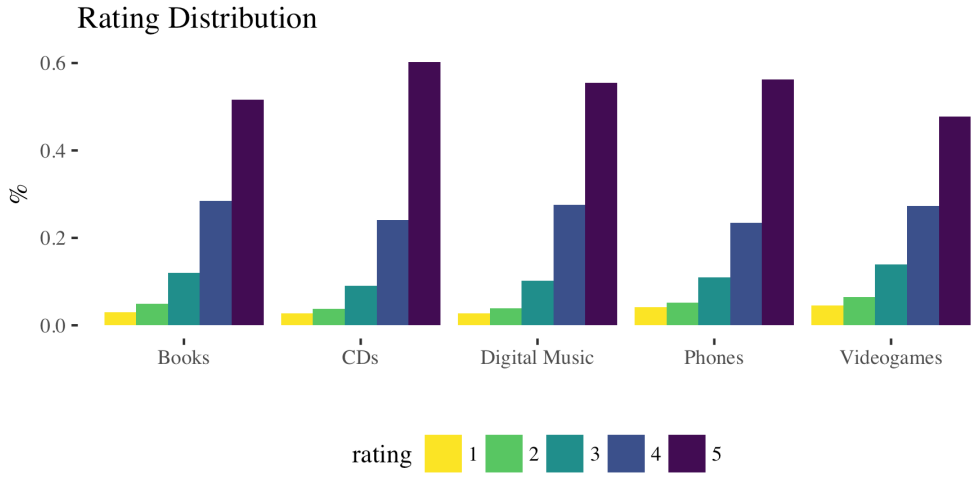

Figure 5.2 Dataset rating distribution for the distinct domains.

We have used these larger datasets to evaluate the aspect-based recommendation methods. Before presenting the obtained experiment results, in the next Section we explain the followed evaluation methodology and used metrics.

5.2 Recommendation methods

Below we present the complete list of recommendation algorithms we have evaluated in this the experiments. Unless stated otherwise they have been implemented on top of RankSys.

- **IPOP**. A popularity-based recommender where the items with a higher number of ratings are recommended to the users. This baseline approach does not consider any personal information.
- **UB-CF.** A user-based collaborative filtering method that exploits the similarities between users to estimate the ratings. We compute the Cosine similarity over the vectors of item ratings and use several numbers of neighbors, namely k=5, 10, 15, 50 and 100.
- **IB-CF.** An item-based collaborative filtering method similar to UB-CF but computing similarities between items. We also compute the Cosine similarity

but the number of neighbors is not fixed and we consider all the items rated by the user.

- **MF.** A matrix-factorization collaborative filtering method. We tested 5, 10, 15, 50 and 100 factors.
- **CB.** A content-based approach that exploits aspects to build the user and item profiles. We test both aspects found with Double Propagation and through LDA, as explained in Section 4.2. We utilize the Cosine similarity between the user and item's vectors.
- **CBCF.** A hybrid recommendation algorithm that combines content-based (CB) as well as collaborative filtering (CF) information. The content part is similar to the content-based only strategy.

5.3 Evaluation methodology and metrics

We test the different pruning strategies on Aspect Extraction methods with Double Propagation by comparing the extracted aspects with the true aspects in the annotated datasets. Annotated aspects correspond to words or set of words that explicitly appear in the texts. We consider that the extracted aspects match the true aspects if they agree on their lemmas. We will also perform a qualitative analysis on the obtained results.

We will also analyze the performance of the recommenders as a ranking problem, as stated in Section 2.2.3 For such purpose, we will follow the *TrainingItems* methodology described in (Bellogin et al., 2011). This strategy uses no information about the ground truth contained in the test set. The training set for each user is computed as the set of items that have been rated for at least one user in the dataset, excluding those pairs (user, item) that belong to the train split. This methodology is more appropriate than the traditional train-test splitting when there are very few ratings for each user, since there shall be very few test ratings for each user. We will follow a 5-fold cross-validation strategy, leaving out 20% of each split for the test set.

In the experiments we will compute the following metrics (see Section 2.2.3 above for the details):

- Precision (P) and Recall (R) to measure the amount of relevant items returned to the user. We will also evaluate these metrics at different cutoffs P@k, R@k, for k=1, 5, 10 and 50.
- Mean Average Precision (MAP) and Normalized Discount Cumulative Gain (NDCG) to measure the quality of the rank in the returned list. We also evaluate cutoff at k=1, 5, 10 and 50.

- User and Item Space Coverage (USC, ISC) to measure the user coverage and item coverage/diversity of the recommenders.

5.4 Results

Double Propagation on annotated Datasets

First, we analyze how different aspect extraction methods behave on annotated datasets.

Table 5.3 shows the number of true, found and correct aspects after running Double Propagation for each dataset, as well as the computed precision and recall values. We show 5 different variations: no pruning, sentence and clause pruning, compound pruning and combinations of sentence/clause with compound pruning. We can see that the highest Recall is obtained when applied only compound pruning. This is the expected behavior since, even though we call it pruning, compound pruning consists on combining extracted aspects to build target sentences, so it expands the aspects found in the previous phases. Higher precision is obtained when the pruning is more intense, i.e. in sentence pruning. The strategy with the highest F1-score depends on the dataset. We can see that Sentence with Compound pruning beats others in two datasets, same as only pruning at the clause level. Clause pruning is the worst at performance time, so we will compare sentence and compound pruning against no pruning in the recommendation models.

Recall that performance metrics are not as good as the reported in (Qiu et al., 2011). Manually exploring annotated datasets we can see that many of the true aspects appear only once in the dataset. These will be discarded because of global pruning removing every target word that does not appear more than once. The manual computation of maximum recall gives smaller values than the reported ones. Maximum precision cannot be computed since it depends on the extraction procedure noise, but it seems that there are some implementation details that are not explicit and modify the performance metrics.

Analyzing our results, we can see that aspects that are not found mostly (~90%) correspond to aspects that appear only once in the dataset, so they are filtered out in the final pruning stage. This rule lowers the recall for these small datasets.

	APEX			CANON			JUKEBOX			NIKON			NOKIA		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
reported	0.87	0.81	0.84	0.90	0.81	0.85	0.90	0.86	0.88	0.81	0.84	0.82	0.92	0.86	0.89
NO	0.16	0.48	0.24	0.11	0.47	0.18	0.13	0.53	0.21	0.10	0.32	0.15	0.18	0.58	0.27
SENTENCE	**0.49**	0.24	0.32	**0.44**	0.21	0.28	**0.40**	0.25	**0.31**	**0.41**	0.16	0.23	**0.49**	0.21	0.29
CLAUSE	0.43	0.31	**0.36**	0.29	0.24	0.26	0.30	0.29	0.29	0.31	0.18	0.23	0.46	0.28	**0.35**
COMPOUND	0.16	**0.65**	0.26	0.13	**0.69**	0.22	0.10	**0.69**	0.17	0.14	**0.57**	0.22	0.19	**0.72**	0.30
SENT+CP	0.23	0.31	0.26	0.25	0.38	**0.30**	0.17	0.39	0.24	0.31	0.35	**0.33**	0.32	0.33	0.32
CLAU+CP	0.22	0.42	0.29	0.20	0.43	0.27	0.15	0.44	0.22	0.25	0.36	0.30	0.31	0.41	**0.35**

Table 5.3 Precision (P), recall (R), and F1 score (F1) of Double Propagation algorithm on the Five product datasets.

Qualitative analysis of Aspect Extraction Methods

We have run the Manual and Double Propagation Developed methods shown in Section 4.1 on the Amazon datasets described above. We keep only those reviews that have been annotated with both methods so the comparison is fair with respect to the Manual extraction methods. Statistics on these filtered datasets are shown in Table 5.4. The distribution of number of ratings per user and per item is also shown in Figure 5.3. We can see that it resembles a log-normal distribution. This pattern is the same as the ratings in the complete dataset.

5core reviews	Books	CDs and Vinyl	Digital music	Phones	Videogames
Number of reviews	160,834	38,966	15,865	66,321	28,405
Number of items	19,704	8,345	3,089	9,141	3,419
Number of users	93,312	19,515	4,012	23,346	12,596

Table 5.4 Statistics on Amazon reviews datasets that have been annotated with aspects and their polarity with Manual and Double Propagation extraction methods.

We analyze the qualitative results of each extraction method to understand their similarities and differences. Recall that the baseline extraction procedure starts from a seed list of words and expand it with synonyms, and Double Propagation with pruning removes the less frequent noun in each sentence and forms compound terms. The top 10 and 50 extracted aspects of each strategy for domain Phones are shown in Table 5.5. We highlight in italics those that Double Propagation is not able to found in the topN

but the guided manual method does; and underscored those aspects that are found with both manual and Double Propagation methods, and. We can see that most of extracted through Double Propagation are not found with the manual. We also highlight the differences between the DP with and without pruning: in red those that are found initially and removed after the pruning phase, and in blue those that are selecting with pruning but were not the most important without pruning. We can see that most of the aspects found with Double Propagation are found either with or without pruning. In the absence of pruning, common words such as "a" or "one" appear in the top found aspects, even thought they are mostly noise. However, they are filtered out when a sentence pruning phase is run. This leads to the identification of a basic aspect as "camera" that is not found without pruning. Compound pruning is able to find aspects made of more than one word that appear very frequently, for example, "battery life" or "screen protector". However, there is still some level of noise even after the pruning stages.

We can see that the average polarity of the aspects extracted in a review correlates with the rating. We show this effect for the analyzed domains in Figure 5.4, where the median of average polarity increases as the review rating does.

Method	Found aspects
Manual	protector *appearance* price screen battery sound buttons speaker size camera *connectivity weight memory microphone processor*
DP Top10	case phone use price screen one time fit look battery
DPP Top10	phone case screen battery price time product charger screen protector iPhone
DP Top50	case phone use price screen one time fit look battery charge product quality charger work button iPhone protector device color thing cover protection back feel review other sound design way day bit power buy problem side a plastic lot port size cable need drop issue volume life love light hand
DPP Top50	phone case screen battery price time product charger screen protector iPhone device quality protector button protection color charge sound use thing cover power review fit battery life ear cable back headset day plastic volume car design port one music problem way bit call work headphone size sound quality look Bluetooth speaker unit camera

Table 5.5 Aspects founds for different aspect extraction procedures. Differences and common aspects are highlighted.

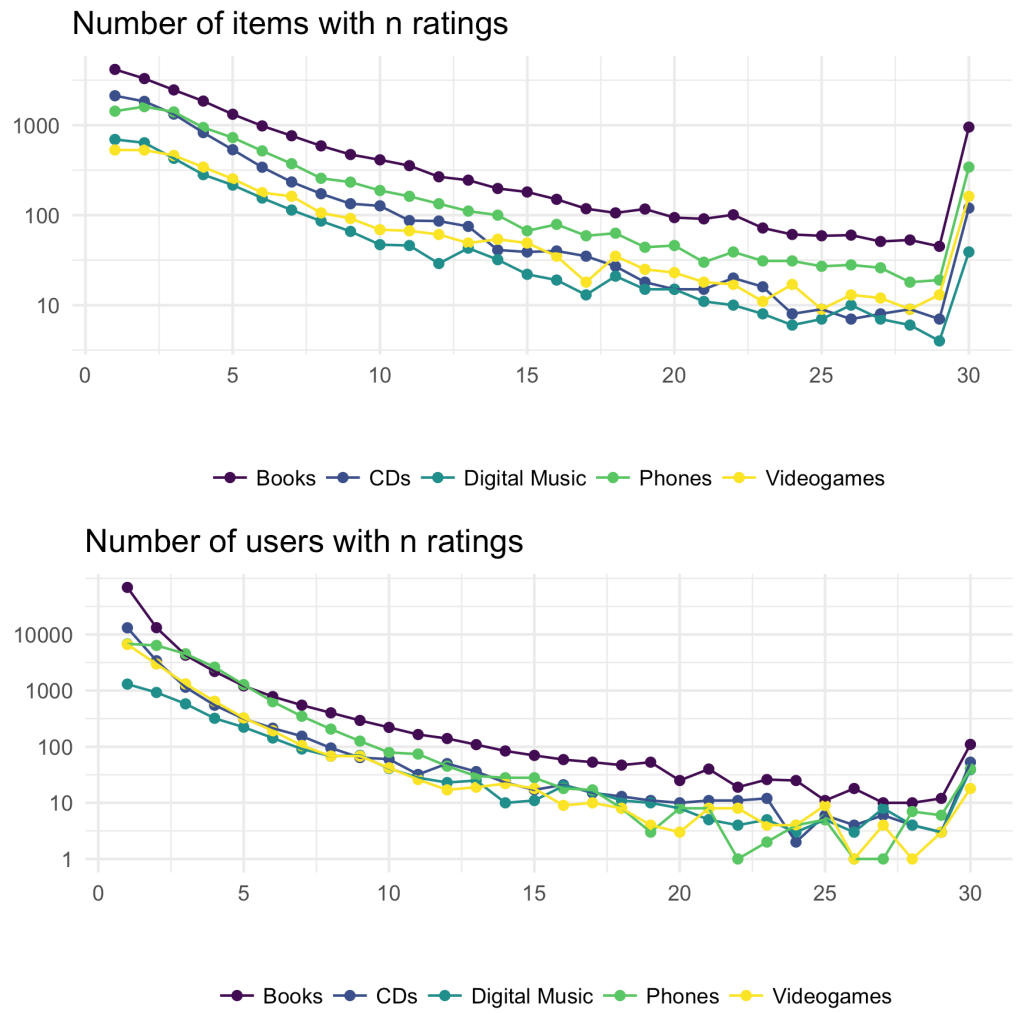

Figure 5.3 Distribution of number of items (top) and users (bottom) with n rating, for the different datasets. Users and items with more than 30 reviews have been collapsed in 30.

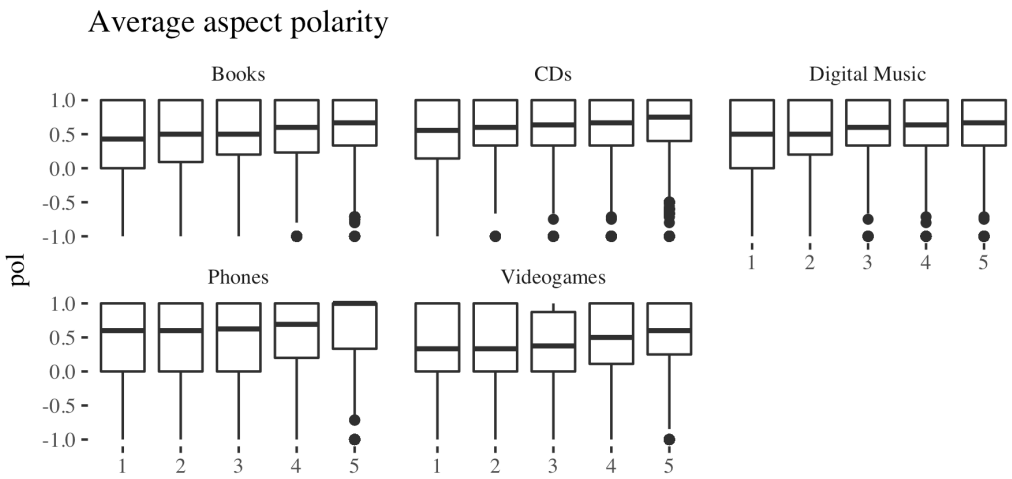

Figure 5.4 Relation between rating review and the average polarity of the aspects extracted in the textual reviews.

Cuantitative analysis of Aspect-Based Recommendation

In this Section we report and analyze the results achieved in the recommendation task with the different methods exposed above, addressing the Research Questions presented in Section 1. The performance metrics of the experiments explained in previous Section are shown in Table 5.6 for the Digital Music domain. We observe very similar behaviors in every domain for most of the metrics and we provide the results of the remaining domains in Appendix A.

Aspect Extraction	Recommender	P@5	P	R@5	R	USC	ISC
-	IPOP	0.003	0.006	0.122	0.022	**1.000**	0.021
	UB50	0.007	0.020	0.221	0.072	0.860	0.449
	UB100	0.008	0.021	0.233	0.076	0.860	0.447
	IB	0.005	0.009	0.144	0.033	0.860	0.451
	MF50	0.006	0.018	0.205	0.064	**1.000**	0.362
	MF100	0.006	0.017	0.185	0.060	**1.000**	0.429
Manual	CB	0.002	0.002	0.046	0.006	0.876	0.459
	CBCF50	0.003	0.006	0.095	0.021	0.876	0.427
DP10	CB	0.001	0.001	0.028	0.004	0.876	0.469
	CBExpl	0.003	0.004	0.109	0.019	0.876	0.470
	CBCF50	0.002	0.003	0.066	0.012	0.876	0.430
DP50	CB	0.001	0.002	0.042	0.006	0.876	0.459
	CBExpl	0.008	0.030	0.328	0.139	0.876	**0.468**
	CBCF50	0.004	0.009	0.139	0.033	0.876	0.416
DP100	CB	0.002	0.002	0.049	0.006	0.876	0.452
	CBExpl	**0.010**	**0.049**	**0.423**	**0.219**	0.876	0.467
	CBCF50	0.004	0.010	0.153	0.038	0.876	0.401
DPP10	CB	0.001	0.002	0.041	0.006	0.876	0.461
	CBExpl	0.003	0.008	0.136	0.034	0.876	0.462
	CBCF50	0.003	0.005	0.093	0.016	0.875	0.427
DPP50	CB	0.002	0.003	0.050	0.008	0.876	0.461
	CBExpl	0.007	0.028	0.290	0.119	0.876	0.467
	CBCF50	0.005	0.010	0.151	0.037	0.875	0.419
DPP100	CB	0.002	0.003	0.057	0.010	0.876	0.459
	CBExpl	0.009	0.040	0.356	0.167	0.876	0.467
	CBCF50	0.005	0.011	0.165	0.042	0.876	0.416
LDA10	CB	0.004	0.006	0.137	0.022	0.876	0.464
	CBCF50	0.006	0.015	0.209	0.052	0.876	0.422
LDA50	CB	0.005	0.014	0.182	0.055	0.876	0.465
	CBCF50	0.008	0.021	0.259	0.078	0.876	0.419

Table 5.6 Recommendation performance values on the Digital Music domain.

We only report P, P@5, R and R@5; other cutoffs behave similar to these metrics. We neither report NDCG and MAP values since they also are correlated to P@5 for every domain and method.

First of all, we observe that Popularity (IPOP) and Matrix Factorization (MF) methods have complete user coverage (USC), a well-know property of these methods that are able to recommend items to any user. In contrast, they achieve some of the lowest item coverage (ISC). IPOP only recommend around 2% of the item set.

Addressing **RQ1**, i.e., determining if the use of information about the users' opinions about item aspects improve the performance of personalized recommendation, we see that it is the case. As shown in the table, the best results are achieved when considering aspects into the recommendation. This observation validates the hypothesis that including aspect information into the recommendation lead to better performance, in terms of several metrics. In particular, ISC is the metric that takes more benefit of considering item aspects.

However, it is important to note that not every aspect-based recommender outperforms the baseline approaches that do not take aspect-based information into account. This important remark needs to be considered when approaching aspect-based recommendation to design a system aimed to benefit from such information.

Regarding **RQ2**, i.e., which aspect extraction strategy generates the most valuable information for recommendation purposes, we see that Double Propagation and LDA are both good approaches and achieve better results than the manual extraction method.

Keeping fixed the number of aspects and using the same recommendation approach, representing items through LDA aspects leads to better results than the corresponding representation with DP aspects. A possible explanation of this result is that LDA provides a complete representation of items, whereas selecting the top N aspects only considers a partial view of the items. This is also one of the main problems of semantic approaches that we have discussed above, i.e., the need for aspect clustering to associate similar or related words that describe the same concept. LDA specifically addresses such problem and we can validate with these results that it is necessary such approach.

Moreover, we can observe that increasing the number of aspects in Double Propagation leads to an improvement on every metric. This supports the idea that there is important information in less popular aspects that should be considered. We expect that increasing the number of aspects we could obtain further improvements. We leave this deeper analysis for future work.

The pruning process in Double Propagation improves the Recall, but gets a lower Precision. There is no much difference on other metrics. This can be somehow related

with the type of aspects each method extracts, as we have seen in the Qualitative Analysis described above. Pruning leads to more curated aspects, whereas in its absence, the many extracted aspects are noise.

Both DP and LDA outperform the manual extraction procedure so we validate that it is worth it to extend human-defined aspects not only because it is done automatically, but also because it leads to more coverage and valuable recommendations.

Considering **RQ3**, i.e., which recommendation technique takes more benefit of aspect-based information, we observe that Content Based, with the user profile explicitly defined from the aspects, is the method that achieves a greater improvement. This strategy is able to represent user's explicit opinion about item aspects that she has evaluated. This seems to be very important information for recommending new items since there are aspects that are shared among items. For instance, if a user finds particularly important the price of a digital camera, it is worth it to consider that information when recommending her mobile phones.

We can also observe that the pure content-based approach is not a good recommender and works worse that most of baselines. This validates the approach of mixing content-based representation with collaborative filtering techniques, as the hybrid method that we are evaluating or the approaches presented in Section 3.2.3 where item representation is included in the latent vectors of Matrix Factorization.

To conclude, we have to say that we have not reported the metric values obtained by the ATagMF algorithm, since they were worse than the baselines for most of the metrics. We think that a possible reason for this low performance is the fact that we considered only aspects with positive polarities, which may imply an important information bias that does not fit the users' preferences. We plan to conduct further exploring this approach in future work, continuation of this thesis.

Chapter 6

Conclusions and future work

In this Chapter we end the thesis summarizing the conclusions derived from the conducted experiments (Section 6.1) and describing several work lines to continue the research presented herein (Section 0).

6.1 Conclusions

In this master thesis we have investigated aspect-based sentiment analysis and recommender systems. We have focused on mining textual reviews which are written by users to describe their satisfaction or opinion about certain items, and which usually have assigned numeric ratings as signals of user preferences. More specifically, we have addressed the task of extracting the item aspects on which the users state their opinions, as well as the associated sentiments. Then we have exploited the extracted aspect-based information for recommendation purposes.

Regarding the *aspect extraction* task, we have developed and evaluated two popular approaches to identify item aspects in textual reviews, namely the *Double Propagation* algorithm, which is based on **semantic relationships** between words to discern which of them are item aspects, and a method that builds a **topic model** inferring latent semantic features, some of them referring to item aspects. We have tested several implementation variations of these approaches, examining the effect of parameter tuning.

To evaluate the extracted aspects and their associated opinion polarity values, we have used them as input data of several *recommender systems*. In particular, we have implemented three recommendation methods that incorporate item aspect information. The first method follows a **content-based** approach where an item is represented as a

vector whose components correspond to the item aspects and whose values are the average opinion polarities extracted from the reviews of the item. We have distinguished between two variations of this method. In a first variation, each user's vector is built as an average aggregation of the assigned polarity to the items aspects the user has rated. In a second variation, a user's vector is a weighted average of the vectors of the items rated by the user, by means of the ratings assigned in the user's reviews.

Our second method follows a **hybrid**, content-based collaborative filtering approach. This method utilizes the same user and item representations that the content-based approach. In traditional heuristic collaborative filtering systems, the similarity between two users (or items) is computed from the ratings that have assigned (or have received). In our aspect-based approach, the similarity between two users (items) is computed with the proposed aspect-based representations.

We have empirically evaluated the previous methods on several relatively large datasets containing item reviews in different domains, namely *books*, *CDs*, *digital music*, *videogames* and *mobile phones*. We have compared the aspect-based methods against various state-of-the-art baselines that do not use aspect opinion information to provide recommendations.

The results achieved in our experiments show that considering aspects significantly improves the quality of recommendations, outperforming state-of-the-art recommenders that do not exploit such information. Specifically, the content-based approach that defines the user and item profiles from the estimated polarity that users assign to the item aspects they explicitly name in the review, is the best strategy from the models we have analyzed. Regarding the aspect extraction procedure, we see that more complete representations of the item profile in terms of its aspect –meaning a holistic representation with topic models or increasing the number of aspect found through DP- also lead to more valuable recommendations for the user.

The study conducted in this thesis thus confirms that user reviews are a very useful source of information, although they are usually omitted in recommendation solutions due to the difficulty of mining textual contents. In this thesis, in contrast, we have shown that exploiting the opinion that users have about specific item aspects has led to significant recommendation improvements consistently in several domains, by means of simple aspect extraction and aspect-based recommendation methods.

6.2 Future work

In this thesis we have presented effective aspect extraction, opinion polarity estimation, and aspect-based recommendation methods. However, we think there are several lines of future work that should be explored in order to improve the results achieved.

In particular, at the aspect extraction stage, we first suggest to analyze different pruning strategies in the Double Propagation algorithm. We have seen that there is room for improvement in such method, since the algorithm wrongly identify aspect words, even performing a significant pruning.

More specifically, we believe that combining the Double Propagation algorithm with the language model approach proposed in (Musto et al., 2017) could lead to a more curated list of relevant aspects. The former exploits syntactic relationships and the latter considers word distributions in review texts, with respect to standard word distributions in English corpora.

We also believe that performing any type of grouping of the found aspects would also help to perform better recommendations. We have found that selecting the most popular from the complete list of found aspects debilitates the performance of the results. A mixed approach between this aspect extraction strategy and the topic model ideas would solve that problem without sacrificing the completeness of the method.

At the recommendation level, we suggest further exploring the ATagMF method presented in this master thesis. We have found that in the current application the method is not able to provide better recommendations than state-of-the-art recommenders that do not use aspect information, but we believe there are a few lines to explore in this approach. For instance, we propose to consider also the aspects with a negative polarity into the model in a fashion that penalizes the estimated item relevance. We will analyze this proposal in future works.

We also suggest extending the Collaborative Topic Rating model presented in Section 3.2.3. This model considers that topic distribution of the reviews and latent factors in the collaborative filtering model are somehow related. We propose to integrate the aspect representation that comes from the semantic-relations approach, instead of the topic models, into the Matrix Factorization model.

Furthermore, we envision that alternative hybrid approaches can be further explored with different user and item representations. In this work we have tested several approaches, namely, topic distribution from LDA, average polarity of aspects from Double Propagation, but there is a wide range of options that could lead to higher improvements.

We also suggest the use of Double Propagation to extract shared aspects between different domains to perform Cross-Domain recommendations. Our analysis suggests

that item coverage is improved when aspect information is considered in the recommendation. That observation, together with the fact that multiple aspects are common to several domains, namely *price*, *weight*, *quality* or *style*, to name a few, suggests us that aspects obtained from item reviews could be a valuable information in this task.

References

Aciar, S., Zhang, D., Simoff, S., Debenham, J., 2007. Informed Recommender: Basing Recommendations on Consumer Product Reviews. IEEE Intelligent Systems 22, 39–47.

Adomavicius, G., Kwon, Y., 2015. Multi-Criteria Recommender Systems, in: Recommender Systems Handbook, Springer, pp. 847–880.

Adomavicius, G., Tuzhilin, A., 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17, 734–749.

Agrawal, R., Srikant, R., 1994. Fast Algorithms for Mining Association Rules in Large Databases, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94. San Francisco, CA, USA, pp. 487–499.

Baeza-Yates, R., Ribeiro-Neto, B., 2011. Modern Information Retrieval: The Concepts and Technology Behind Search. Addison Wesley.

Bauman, K., Liu, B., Tuzhilin, A., 2016. Recommending items with conditions enhancing user experiences based on sentiment analysis of reviews. in: Proceedings of the 3rd Workshop on New Trends in Content-Based Recommender Systems 1673, 19–22.

Bellogín, A., 2012. Performance prediction and evaluation in recommender systems: An information retrieval perspective. Universidad Autónoma de Madrid.

Bellogin, A., Castells, P., Cantador, I., 2011. Precision-oriented Evaluation of Recommender Systems: An Algorithmic Comparison, in: Proceedings of the 5th ACM Conference on Recommender Systems, RecSys '11. ACM, New York, NY, USA, pp. 333–336.

Bennett, J., Lanning, S., Netflix, N., 2007. The Netflix Prize, in: Proceedings of KDD Cup and Workshop in Conjunction with KDD. San Jose, California.

Berge, J.M.F. ten, 1993. Least Squares Optimization in Multivariate Analysis. DSWO Press, Leiden University.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022.

Caputo, A., Basile, P., Gemmis, M. de, Lops, P., Semeraro, G., Rossiello, G., 2017. SABRE: A Sentiment Aspect-Based Retrieval Engine, in: Information Filtering and Retrieval: DART 2014: Revised and Invited Papers, Studies in Computational Intelligence, pp. 63–78.

Carenini, G., Ng, R.T., Zwart, E., 2005. Extracting Knowledge from Evaluative Text, in: Proceedings of the 3rd International Conference on Knowledge Capture, K-CAP '05, pp. 11–18.

Crammer, K., Singer, Y., 2001. Pranking with Ranking, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, NIPS'01, pp. 641–647.

Diao, Q., Qiu, M., Wu, C.-Y., Smola, A.J., Jiang, J., Wang, C., 2014. Jointly Modeling Aspects, Ratings and Sentiments for Movie Recommendation (JMARS), in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, pp. 193–202.

Enrich, M., Braunhofer, M., Ricci, F., 2013. Cold-Start Management with Cross-Domain Collaborative Filtering and Tags. in: Proceedings of the 2013 International Conference on Electronic Commerce and Web Technologies, pp. 101–112.

Esuli, A., Sebastiani, F., 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining, in: In Proceedings of the 5th Conference on Language Resources and Evaluation, LREC '06. pp. 417–422.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D.S., Yates, A., 2005. Unsupervised named-entity extraction from the Web: An experimental study. Artificial Intelligence 165, 91–134.

Fernández-Tobías, I., 2017. Matrix factorization models for cross-domain recommendation: Addressing the cold start in collaborative filtering. PhD thesis. Universidad Autónoma de Madrid.

Ganu, G., Elhadad, N., Marian, A., 2009. Beyond the Stars: Improving Rating Predictions using Review Text Content, in: Proceedings of the 12th International Workshop on the Web and Databases, WebDB 2009, Providence, Rhode Island, USA.

Golder, S.A., Huberman, B.A., 2006. Usage Patterns of Collaborative Tagging Systems. Journal of Information Science 32, 198–208.

Gomez-Uribe, C.A., Hunt, N., 2015. The Netflix Recommender System: Algorithms, Business Value, and Innovation. ACM Transactions on Management Information Systems 6, 13:1–13:19.

Herlocker, J.L., Konstan, J.A., Riedl, J., 2002. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. Information Retrieval 5, 287–310.

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T., 2004. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems 22, 5–53.

Hofmann, T., 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning, AAIM '08 42, 177–196.

Hu, M., Liu, B., 2004a. Mining Opinion Features in Customer Reviews, in: Proceedings of the 19th National Conference on Artifical Intelligence, AAAI '04, San Jose, California, pp. 755–760.

Hu, M., Liu, B., 2004b. Mining and Summarizing Customer Reviews, in: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 168–177.

Hu, Y., Koren, Y., Volinsky, C., 2008. Collaborative Filtering for Implicit Feedback Datasets, in: Proceedings of the Eighth IEEE International Conference on Data Mining, ICDM '08, Pisa, Italy, pp. 263–272.

Hummel, R.A., Zucker, S.W., 1983. On the foundations of relaxation labeling processes. IEEE Transactions on Pattern Analysis and Machine Intelligence 5, 267–287.

Jakob, N., Gurevych, I., 2010. Extracting Opinion Targets in a Single- and Cross-domain Setting with Conditional Random Fields, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pp. 1035–1045.

Järvelin, K., Kekäläinen, J., 2002. Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Information Systems 20, 422–446.

Jin, W., Ho, H.H., 2009. A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, pp. 465–472.

Johnson, C., 2014. Algorithmic Music Recommendations at Spotify.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix Factorization Techniques for Recommender Systems. Computer 42, 30–37.

Lafferty, J.D., McCallum, A., Pereira, F.C.N., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in: Proceedings of the 18th International Conference on Machine Learning, ICML '01, San Francisco, CA, USA, pp. 282–289.

Linden, G., Smith, B., York, J., 2003. Amazon.Com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing 7, 76–80.

Liu, B., 2012. Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

Manzato, M.G., 2013. gSVD++: Supporting Implicit Feedback on Recommender Systems with Metadata Awareness, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, pp. 908–913.

McAuley, J., Leskovec, J., 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text, in: Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pp. 165–172.

McAuley, J., Leskovec, J., Jurafsky, D., 2012. Learning Attitudes and Attributes from Multi-aspect Reviews, in: Proceedings of the 12th International Conference on Data Mining, ICDM '12, pp. 1020–1025.

Miller, G.A., 1995. WordNet: A Lexical Database for English. Communications of the ACM 38, 39–41.

Musto, C., de Gemmis, M., Semeraro, G., Lops, P., 2017. A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews, in: Proceedings of the 11th ACM Conference on Recommender Systems, RecSys '17, pp. 321–325.

Nichols, D., 1998. Implicit Rating and Filtering, in: Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering. ERCIM, Budapest, Hungary, pp. 31–36.

Oard, D., Kim, J., 1998. Implicit Feedback for Recommender Systems, in: Proceedings of the AAAI Workshop on Recommender Systems. Madison, WI, pp. 81–83.

Popescu, A.-M., Etzioni, O., 2005. Extracting Product Features and Opinions from Reviews, in: Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, pp. 339–346.

Poria, S., Cambria, E., Gui, C., Gelbukh, A., 2014. A Rule-Based Approach to Aspect Extraction from Product Reviews, in: Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP). Dublin, Ireland, pp. 28–37.

Qiu, G., Liu, B., Bu, J., Chen, C., 2011. Opinion Word Expansion and Target Extraction Through Double Propagation. Computational Linguistics 37, 9–27.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. in: Proceedings of the IEEE 77(2), 257–286.

Robbins, H., Monro, S., 1951. A Stochastic Approximation Method. The Annals of Mathematical Statistics 22, 400–407.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2001. Item-based Collaborative Filtering Recommendation Algorithms, in: Proceedings of the 10th International Conference on World Wide Web, WWW '01, pp. 285–295.

Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., Jin, C., 2007. Red Opal: Product-feature Scoring from Reviews, in: Proceedings of the 8th ACM Conference on Electronic Commerce, EC '07, pp. 182–191.

Schuster, S., Manning, C.D., 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks, in: Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC '16.

Shardanand, U., Maes, P., 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth." ACM Press, pp. 210–217.

Titov, I., McDonald, R., 2008a. Modeling Online Reviews with Multi-grain Topic Models, in: Proceedings of the 17th International Conference on World Wide Web, WWW '08, Beijing, China, pp. 111–120.

Titov, I., McDonald, R.T., 2008b. A Joint Model of Text and Aspect Ratings for Sentiment Summarization., in: Proceedings of ACL-08: HLT. Association for Computational Linguistics, Columbus, Ohio, pp. 308–316.

Wang, C., Blei, D.M., 2011. Collaborative Topic Modeling for Recommending Scientific Articles, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, pp. 448–456.

Wang, H., Lu, Y., Zhai, C., 2010. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, pp. 783–792.

Wang, Y., Liu, Y., Yu, X., 2012. Collaborative Filtering with Aspect-Based Opinion Mining: A Tensor Factorization Approach, in: Proceedings of the 12th International Conference on Data Mining, pp. 1152–1157.

Wu, Y., Ester, M., 2015. FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, pp. 199–208.

Zhao, W.X., Jiang, J., Yan, H., Li, X., 2010. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pp. 56–65.

Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R., 2008. Large-Scale Parallel Collaborative Filtering for the Netflix Prize, in: Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management, AAIM '08, pp. 337–348.

Zhuang, L., Jing, F., Zhu, X.-Y., 2006. Movie Review Mining and Summarization, in: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM '06, pp. 43–50.

Appendix A: Experimental results

In this appendix we present extended results for the offline experiments that we have conducted in this master thesis to evaluate the aspect extraction and aspect-based recommendation methods.

Extracted aspects from textual reviews

Next we show the aspects identified by the Manual and Double Propagation methods for different domains. We observe that manual approach finds really good aspects (in terms of correctness) but is not able to reach a high coverage. Performing pruning stages lead to more specific aspects and remove words that are not meaningful for describing an item and are mostly noise.

Method	Extracted aspects - BOOKS
Manual	story characters literary style ending script descriptions pictures price pacing coherence
DP Top10	book story read character love one time author end way
DPP Top10	book story character time author life way love series read
DP Top50	book story read character love one time author end way life thing series other plot novel reader people work year man world part lot page review bit a friend woman write something family look romance start feel day place point job line relationship this action word scene fact while child
DPP Top50	book story character time author life way love series read plot novel people end thing world reader man romance family something lot main character review work woman one page mystery bit part friend other story line year reading action first book girl love story fun relationship next book history great story child day good story heart point

Method	Extracted aspects - CDS
Manual	timbre sounds lyrics rhythm melody style harmony picture price story ending cast characters art style texture dynamics costumes
DP Top10	song album sound music one time track voice lyric CD
DPP Top10	album song music sound track time band voice CD guitar
DP Top50	song album sound music one time track voice lyric CD love band fan record way work year a guitar vocal thing other rock release day style something end version melody hit beat lot people favorite all part bit feel cd play show this pop performance singer solo tune set quality
DPP Top50	album song music sound track time band voice CD guitar rock way love work fan cd something thing version beat record vocal people one pop metal performance release year best song great song production title track solo hit show set other good song collection day disc John singing style artist end favorite song lyric singer

Method	Extracted aspects – GAMES
Manual	graphics story controls characters gameplay sounds music price script ending difficulty art style customization pacing
DP Top10	game play graphic time one fun story character thing way
DPP Top10	game time fun story character thing way gameplay level play
DP Top50	game play graphic time one fun story character thing way lot gameplay control a other level use look people system feel player fan end bit enemy point sound hour part series weapon fight something year music while review experience work mode world move action buy problem price battle voice everything
DPP Top50	game time fun story character thing way gameplay level play people lot enemy system control player weapon mode one series world graphic something music controller great game mission sound video game action voice battle point PS3 part combat other price game play end other game everything bit version review Wii button experience multiplayer gun

Method	Extracted aspects - MUSIC
Manual	lyrics timbre sounds rhythm melody style harmony ending price texture dynamics
DP Top10	album song sound music track lyric one time love voice
DPP Top10	album song music track sound time band voice beat CD
DP Top50	album song sound music track lyric one time love voice way year beat vocal fan band CD thing work release record guitar other rock hit style melody day something pop people feel lot a classic end artist bit title production favorite ballad man part tune cd debut rap all version
DPP Top50	album song music track sound time band voice beat CD way guitar rock love cd work record pop rap title track thing vocal people fan one hit something best song production year release great song artist lyric good song version best album debut day piano style debut album man other lot ballad bit favorite song chorus first album

Recommendation Methods Results

Next we show the performance metrics of offline experiments carried on to address the problem of aspect-based recommendation on several domains. The conclusions derived from them are equivalent to those exposed in Section 5.4, analyzing the Digital Music domain.

Aspect Extraction	Recommender	P@5	P	R@5	R	USC	ISC
-	IPOP	0.003	0.005	0.107	0.021	**1.000**	0.008
	UB50	0.007	0.015	0.134	0.058	0.531	0.353
	UB100	0.006	0.018	0.163	0.069	0.552	0.355
	IB	0.004	0.007	0.111	0.025	0.553	0.352
	MF50	0.003	0.010	0.106	0.037	**1.000**	0.189
	MF100	0.003	0.009	0.101	0.037	**1.000**	0.265
Manual	CB	0.001	0.001	0.029	0.006	0.586	0.407
	CBCF50	0.003	0.007	0.076	0.022	0.581	0.410
DP10	CB	0.000	0.001	0.016	0.002	0.586	0.421
	CBExpl	0.002	0.004	0.085	0.017	0.586	0.421
	CBCF50	0.001	0.002	0.033	0.008	0.582	0.406
DP50	CB	0.001	0.001	0.024	0.003	0.586	0.414
	CBExpl	0.007	0.033	0.318	0.159	0.586	**0.422**
	CBCF50	0.003	0.008	0.102	0.029	0.581	0.402
DP100	CB	0.001	0.001	0.027	0.004	0.586	0.405
	CBExpl	**0.009**	**0.053**	**0.424**	**0.250**	0.586	0.420
	CBCF50	0.003	0.009	0.114	0.034	0.581	0.394
DPP10	CB	0.001	0.001	0.024	0.004	0.586	0.384
	CBExpl	0.002	0.006	0.112	0.026	0.586	0.383
	CBCF50	0.002	0.004	0.050	0.012	0.579	0.404
DPP50	CB	0.001	0.002	0.035	0.007	0.586	0.410
	CBExpl	0.006	0.025	0.265	0.117	0.586	0.417
	CBCF50	0.003	0.010	0.110	0.034	0.580	0.404
DPP100	CB	0.001	0.003	0.051	0.010	0.586	0.409
	CBExpl	0.008	0.037	0.336	0.168	0.586	0.419
	CBCF50	0.004	0.011	0.123	0.038	0.580	0.403
LDA10	CB	0.003	0.005	0.110	0.020	0.586	0.420
	CBCF50	0.006	0.010	0.134	0.040	0.582	0.405
LDA50	CB	0.005	0.012	0.176	0.048	0.586	0.418
	CBCF50	0.009	0.017	0.181	0.065	0.582	0.399

Table A.1 Recommendation performance values on the CDs domain.

Aspect Extraction	Recommender	P@5	P	R@5	R	USC	ISC
-	IPOP	0.004	0.009	0.170	0.039	**1.000**	0.020
	UB50	0.007	0.015	0.168	0.064	0.684	0.520
	UB100	0.006	0.017	0.207	0.075	0.707	0.520
	IB	0.004	0.005	0.129	0.023	0.710	0.522
	MF50	0.004	0.013	0.171	0.055	**1.000**	0.304
	MF100	0.004	0.011	0.154	0.048	**1.000**	0.416
Manual	CB	0.001	0.002	0.052	0.009	0.725	0.558
	CBCF50	0.003	0.006	0.092	0.022	0.719	0.557
DP10	CB	0.001	0.001	0.026	0.003	0.725	0.562
	CBExpl	0.003	0.006	0.155	0.031	0.725	0.562
	CBCF50	0.002	0.003	0.060	0.012	0.718	0.556
DP50	CB	0.001	0.001	0.031	0.004	0.725	0.566
	CBExpl	0.008	0.040	0.381	0.190	0.725	**0.566**
	CBCF50	0.004	0.008	0.131	0.032	0.716	0.556
DP100	CB	0.001	0.001	0.035	0.004	0.725	0.565
	CBExpl	**0.011**	**0.061**	**0.493**	**0.290**	0.725	0.566
	CBCF50	0.004	0.009	0.150	0.039	0.716	0.553
DPP10	CB	0.001	0.001	0.031	0.005	0.725	0.505
	CBExpl	0.004	0.009	0.164	0.041	0.725	0.505
	CBCF50	0.002	0.003	0.068	0.012	0.715	0.549
DPP50	CB	0.001	0.002	0.036	0.006	0.725	0.554
	CBExpl	0.008	0.035	0.344	0.164	0.725	0.555
	CBCF50	0.003	0.007	0.122	0.030	0.716	0.557
DPP100	CB	0.001	0.002	0.046	0.008	0.725	0.558
	CBExpl	0.009	0.048	0.417	0.224	0.725	0.560
	CBCF50	0.004	0.009	0.135	0.036	0.716	0.555
LDA10	CB	0.002	0.003	0.069	0.013	0.725	0.563
	CBCF50	0.004	0.008	0.126	0.034	0.718	0.546
LDA50	CB	0.003	0.007	0.122	0.031	0.725	0.564
	CBCF50	0.007	0.014	0.175	0.055	0.717	0.546

Table A.2 Recommendation performance values on the Videogames domain.

Aspect Extraction	Recommender	P@5	P	R@5	R	USC	ISC
-	IPOP	0.002	0.004	0.108	0.020	**1.000**	0.006
	UB50	0.004	0.009	0.102	0.040	0.866	0.512
	UB100	0.004	0.012	0.122	0.049	0.871	0.511
	IB	0.003	0.005	0.088	0.020	0.871	0.514
	MF50	0.003	0.010	0.132	0.042	**1.000**	0.145
	MF100	0.003	0.009	0.124	0.038	**1.000**	0.237
Manual	CB	0.001	0.001	0.023	0.004	0.884	0.486
	CBCF50	0.001	0.004	0.052	0.016	0.883	0.523
DP10	CB	0.000	0.000	0.013	0.002	0.884	0.481
	CBExpl	0.002	0.003	0.073	0.016	0.884	0.481
	CBCF	0.001	0.002	0.034	0.008	0.881	0.523
DP50	CB	0.001	0.001	0.021	0.003	0.884	0.533
	CBExpl	0.004	0.017	0.206	0.083	0.884	**0.535**
	CBCF	0.002	0.005	0.071	0.020	0.883	0.523
DP100	CB	0.001	0.001	0.027	0.004	0.884	0.531
	CBExpl	**0.006**	**0.028**	**0.275**	**0.133**	0.884	0.536
	CBCF	0.002	0.006	0.079	0.023	0.883	0.524
DPP10	CB	0.000	0.001	0.019	0.003	0.884	0.388
	CBExpl	0.002	0.004	0.076	0.017	0.884	0.387
	CBCF	0.001	0.003	0.040	0.010	0.879	0.516
DPP50	CB	0.001	0.001	0.023	0.004	0.884	0.485
	CBExpl	0.004	0.013	0.160	0.063	0.884	0.487
	CBCF	0.002	0.005	0.068	0.019	0.882	0.525
DPP100	CB	0.001	0.001	0.025	0.004	0.884	0.500
	CBExpl	0.004	0.019	0.197	0.086	0.884	0.504
	CBCF	0.002	0.006	0.074	0.022	0.883	0.526
LDA10	CB	0.001	0.001	0.026	0.003	0.884	0.536
	CBCF	0.002	0.005	0.070	0.020	0.883	0.509
LDA50	CB	0.002	0.003	0.064	0.012	0.884	0.534
	CBCF	0.003	0.008	0.103	0.034	0.883	0.514

Table A.3 Recommendation performance values on the Phones domain.

Aspect Extraction	Recommender	P@5	P	R@5	R	USC	ISC
	IPOP	0.002	0.005	0.091	0.022	**1.000**	0.003
	UB50	0.008	0.013	0.115	0.057	0.401	0.424
	UB100	0.006	0.015	0.137	0.065	0.451	0.428
	IB	0.004	0.007	0.105	0.032	0.489	0.431
	MF50	0.002	0.007	0.080	0.026	**1.000**	0.069
	MF100	0.002	0.007	0.082	0.029	**1.000**	0.125
Manual	CB	0.000	0.001	0.013	0.002	0.507	0.455
	CBCF50	0.001	0.002	0.023	0.007	0.483	0.484
DP10	CB	0.000	0.000	0.007	0.000	0.507	0.485
	CBExpl	0.001	0.001	0.040	0.005	0.507	0.486
	CBCF	0.001	0.001	0.016	0.004	0.488	0.491
DP50	CB	0.000	0.000	0.009	0.002	0.507	0.506
	CBExpl	0.004	0.016	0.172	0.076	0.507	**0.510**
	CBCF	0.002	0.005	0.069	0.022	0.477	0.499
DP100	CB	0.000	0.000	0.011	0.002	0.507	0.502
	CBExpl	**0.005**	**0.028**	**0.250**	**0.135**	0.507	0.509
	CBCF	0.003	0.008	0.082	0.031	0.474	0.493
DPP10	CB	0.000	0.000	0.013	0.001	0.507	0.422
	CBExpl	0.001	0.002	0.050	0.010	0.507	0.422
	CBCF	0.001	0.001	0.021	0.005	0.497	0.473
DPP50	CB	0.000	0.001	0.015	0.003	0.507	0.480
	CBExpl	0.003	0.013	0.146	0.061	0.507	0.484
	CBCF	0.002	0.006	0.068	0.023	0.479	0.491
DPP100	CB	0.000	0.001	0.019	0.003	0.507	0.480
	CBExpl	0.005	0.021	0.201	0.096	0.507	0.492
	CBCF	0.003	0.008	0.080	0.030	0.477	0.491
LDA10	CB	0.001	0.002	0.055	0.010	0.507	0.510
	CBCF	0.004	0.006	0.089	0.025	0.482	0.502
LDA50	CB	0.002	0.006	0.082	0.026	0.507	0.503
	CBCF	0.007	0.012	0.128	0.048	0.481	0.488

Table A.4 Recommendation performance values on the Book domain.